Flow Anomaly Based Intrusion Detection System for Android Mobile Devices

Panagiotis I. Radoglou-Grammatikis, Panagiotis G. Sarigiannidis
Department of Informatics and Telecommunications Engineering
University of Western Macedonia
Kozani, Greece
pradoglou@uowm.gr, psarigiannidis@uowm.gr

***Abstract*—The penetration of the modern mobile devices is progressively gaining ground in today's cognitive applications and services. Several applications have become part of the smartphone capabilities such as e-mail monitoring, Internet browsing, social networks activities, etc. However, the increased computation and storage capabilities of smartphones have attracted more and more cyber attacks in terms of writing mobile malware for various purposes. In this paper, we present an intrusion detection system (IDS) for detecting the anomaly behaviors in Android mobile devices. The IDS continuously monitors the network traffic of the mobile device and collects various features of the NetFlows. An artificial neural network (ANN) gathers the data flows and determines whether there is an invasion or not. The proposed IDS is demonstrated in realistic conditions, where the accuracy of the systems reaches 85%.**

Keywords—Intrusion Detection System, Security, NetFlows, Mobile, Android, Artificial Neural Networks

I. INTRODUCTION

Over the last few years, mobile devices have gained increasing popularity due to the variety of the data services they offer, such as browsing the Internet, e-mailing, gaming, along with the traditional voice services. Nowadays, such devices are equipped with enough facilities to even replace the usage of laptops. Since the appearance of the Android and iOS operating systems, mobile devices have changed dramatically concerning hardware, software, and user interface capabilities, leading to a new era of smartphones. These devices, are getting constantly smaller, cost-effective, more convenient and powerful. In addition, they are now able to provide a plethora of advanced data input interfaces, enabling users to effectively interact with the device. Typical examples of such advanced features include software keyboards displayed on a touchscreen instead of hard ones, magnetometer, and gyroscope for measuring or maintaining the orientation of the device.

However, the expansion of the mobile devices comes with the increase of size of information that is processed, resulting in increasing the potential targets stemming from an ever-increasing number of various security threats. Modern mobile applications are open and programmable [1,2]. As a result, they are getting more vulnerable to various malicious attacks, such as Trojan horses, worms, and mobile botnets [3]. According to the Google Play and Apple App Store, Android and iOS are the two most popular operating systems sharing together a percent of over 70% of the total market. But it is also estimated that almost ten million Android smartphones have been affected by some malware in 2016, while the 33,9% of the free iOS applications experienced some kind of hidden capability with the intent to leak private user information to the public. Therefore, despite the number of the security techniques that modern operating mobile computing systems incorporate, such as access control and (post) authentication techniques, it is necessary to examine new solutions which would protect the mobile devices from unknown and undefined threats. Also, the new security mechanisms have to be designed suitable in order to apply to the mobile devices, since they haven't equivalent resources, such as the conventional computing systems.

A wide range of security components appears in the market in response to security issue of mobile computing systems. Most of the mobile security software focus on the traditional PC security-based approaches. However, the considerable limitations in terms of CPU, memory and battery power pose limitations in adopting these solutions. In the light of the aforementioned remarks, this work endeavors to identify a light-weight, scalable, and efficient intrusion detection system for Android environment. These mechanisms provide services for addressing well-known attacks, but also include techniques for identifying unknown threats intrusions. The proposed solution intends to examine specific features of the bidirectional NetFlows and analyzes these data streams using effective machine learning algorithms. Specifically, we obtained malicious and normal IP NetFlows under controlled experiments in order to train an artificial neural network which will be able to detect abnormal behaviors in Android devices.

The remainder of the paper is organized as follows: In Section II we discuss the related intrusion detection technologies for mobile devices. Section III describes the architecture and the design of the proposed IDS. Section IV evaluates the introduced IDS and Section V gives the concluding remarks of our work.

II. RELATED WORK

Security aspects in mobile devices have already been well-studied, giving emphasis to the development of new and effective IDS. Anomaly-based and signature-enabled method and strategies have been developed for collecting malware features and then identifying malware software or actions.

Schmidt et al. [4] proposed a solution based on monitoring the events occurring on Linux-Kernel level. The proposed monitoring module collects features like CPU usage, RAM

978-1-5090-4386-6/17/$31.00 ©2017 IEEE

usage, and kernel system calls. However, at that (publication) time, no physical Android devices were available. Hence, the developed systems were compatible only with Symbian OS.

Shabtai et al. [7] presented a host based framework for detecting malware on Android devices. The proposed system continuously monitors various features and events of smartphones and then apply machine learning mechanisms to classify the collected information. However, the proposed system was not assessed with real malware.

Yuan et al. [5] proposed an IDS for anomaly detection on Android smartphones. The introduced framework continuously monitored the information obtained from a smartphone and then it applied the Naive Bayes Algorithm to classify the collected data as normal or malicious. However, the claims regarding the detection rates remain questionable since the malware, which is employed in the training of the Naive Bayes Classifier aren't mentioned.

Also, in [6] a signature-based IDS for Android devices was presented. The authors used the signature-based detection technique with predefined rules for detecting anomalies. However, this method is limited to well-known threats, neglecting thus unknown threats. It is worth mentioning that given the high-level of security threats, it is necessary to explore new solutions, capable of protecting the modern mobile devices from both known and undefined threats.

Most of the methods mentioned in the literature are derived from PC anomaly detection techniques and don't take into account the unique characteristics of the mobile devices. A new kind of IDS for mobile devices is proposed which takes into consideration the CPU, the memory and the power consumption of the mobile device. The detection in the proposed system is achieved by continual monitoring of the NetFlows. Specifically, the analyzing process of the NetFlows is performed locally on the mobile devices and in this way, our approach overcomes the privacy issue. To this end, a powerful Python module is introduced for capturing and analyzing the ongoing network traffic. The rationale behind selecting Python is to enhance our proposed IDS with flexibility since it is able to operate in many system architectures such as Linux, Windows, Android and other operating systems.

III. PROPOSED ANDROID-ENABLED IDS

The main goal of an effective IDS is to provide high rates of attack detection with very small rates of false alarms. The key components of these systems are [8]:

- Information Source: Data utilized by the IDS.
- Analysis Engine: Process by which the intrusion detection is made.
- Response: Action taken when an intrusion is detected.

The principal module is the analysis engine. The analysis engine applies three types of techniques for analyzing and detecting a security threat. These methods are the signature-based technique, the anomaly detection, and the protocol anomaly detection. In our approach, the second method is adopted, which usually depends on a machine learning algorithm. The machine learning algorithm is used for detecting unknown threats. For this purpose, it is prior trained with the aim of normal and malicious traces. In the context of this paper an artificial neural network due to its light-weight operation and its efficiency.

An ANN is an information-processing model which consists of individual neurons. Each neuron is fundamentally a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as an input to all of the neurons in the next layer. During training, the neural network parameters are optimized to associate outputs (each output represents a class of a NetFlow, like normal behavior and attack) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the NetFlow). When an ANN is applied, it processes the input patterns and tries to output the corresponding class. Indicative efforts on the intrusions detection using artificial neural networks have been presented in [10,11]. In our approach, we utilized a multi-layer Perceptron (MLP). MLP is a layered feed forward ANN which typically trained with back propagation algorithms. These networks have implemented into countless applications requiring static pattern classification. Their main advantage is that they are easy to use and that they can approximate any input/output map.

While anomaly detection systems are conceptually attractive, many design and development challenges remain to be addressed. These challenges include the remedy of the high false alarm rate phenomena and the need for gigabit speeds scaling up, just to mention some. The flow-based anomaly detection method is one of those approaches that rely on aggregated traffic metrics. Its main merit lies in the host independence and the usability on high-speed networks. Also, the flow-based techniques can be used to detect a variety of threats, such as scan attacks, worms, botnets and denial of service (DoS) attacks [9].

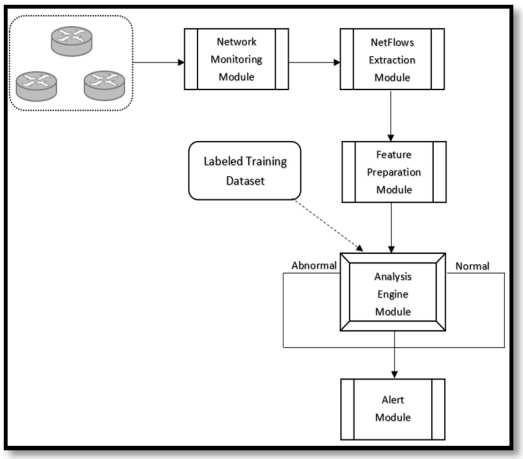

Figure 1: The proposed IDS architecture.

Fig. 1 presents the main modules of the proposed IDS. It is composed of five main modules. The first component is the network monitoring module which captures and analyses the network traffic. Then, the flow extraction module extracts the corresponding NetFlows from the network information and the feature extraction module extracts specific features from the NetFlows, which are used to detect the abnormal behaviors. The analysis engine module utilizes the aforementioned ANN

to analyze the extracted features, by determining whether there is an invasion or an abnormal behavior. Finally, the alert module produces information about the NetFlows, when presenting abnormal behavior. In the following, each one of the main modules is described in detail.

A. Network Monitoring Module

The main operation of this module is to monitor the network information. When the user enables the operation of this module is created an Android service, which continually captures and analyses the network traffic. To this end, we utilized the Scapy library. Scapy is a packet manipulation library for computer networks written in Python. It can forge or decode packets, send them on the wire, capture them, and match requests and replies.

B. Flows Extraction Module

The principal operation of this module is to collect the flow data exported from the network traffic. When the user terminates the network monitoring procedure, the network information is sent to the flow extraction module, which extracts from them the corresponding bidirectional NetFlows. Note that all this information is stored locally in the mobile device.

C. Feature Extraction Module

The feature extraction module receives the flow data and exports specific features which are important to detect abnormal behaviors. The ANN of the analysis engine is trained with similar features so as to be able to determine whether there is an invasion or an abnormal behavior. We selected the following features of a NetFlow information:

- Average NetFlow Size: It provides a useful hint for anomalous events, such as port scan. It is typically very small in order to increase the efficiency of attacks.
- Average Packet Number: One of the main features of DoS attacks is the source IP spoofing, which makes the task of tracing attacker's true source very difficult. A side effect is the generation of flows with a small number of packets, i.e., about 3 packets per flow. This differs from normal traffic that usually involves a higher number of packets per flow.
- Average Packet Size: Another factor is the size of each packet in the flow. Low average size can be a sign of anomaly. For example, in TCP flooding attacks packets of 120 bytes are typically sent.
- NetFlow Duration: The duration of a NetFlow can indicate many kinds of intrusions, such as worms, Botnets, and DoS attacks.

D. Analysis Engine Module

The analysis engine is the core module of the IDS. An MLP network was adopted as the classifier which classifies the corresponding NetFlows. Fig. 2 presents the architecture of our proposed ANN. The input layer of the MLP network contains four neurons for the corresponding features of the NetFlows. A hidden layer is defined which includes 85 neurons. Each neuron of the hidden layer applies the hyperbolic tangent sigmoid transfer function. Finally, a neuron is used in the output layer which utilizes the log-sigmoid transfer function. If the output of the ANN is equal or bigger than 0,5, then the corresponding NetFlow is classified as abnormal, otherwise as normal. For the training process, we utilized the Levenberg-Marquardt algorithm and 145438 NetFlows from our experiments and from the CTU-13 dataset [12].

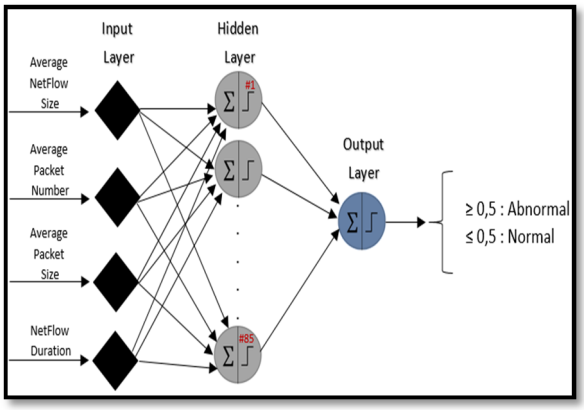

Figure 2: Architecture of proposed ANN.

E. Alert Module

This module runs at the final stage of the proposed system. It demonstrates the characteristics of the NetFlows which exhibited abnormal behavior.

IV. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed IDS in an Android platform. For the integration of the experimental measurements, we used the Sony Xperia P (LT22i) smartphone with a 2x ARM-Cortex-A9 processor at 1.00 GHz with Android version 4.1.2. and Huawei Ascend p8 Lite with an 8x ARM-Cortex-A53 at 1.2 GHz with Android version 5.0.1. A set of 70 and 30 applications were collected with normal and malicious behavior respectively for the needs of the experiments. For each experiment, we utilized 12 and 8 random selections with normal and malicious behavior respectively.

The accuracy of the proposed IDS was assessed based on the following methodology. When an intrusion is indicated correctly, we have a "True Positive" (TP) fact. When a non-intrusion is indicated and this assertion is correct, we have a "True Negative" (TN). When the IDS indicates an intrusion and this assertion is wrong a "False Positive" (FP) alarm is triggered. Lastly, when a non-intrusion is indicated and an intrusion is indeed in progress, we have a "False Negative" (FN) incident. FN is the worst case situation of every detection mechanism since it causes a false alarm. Given these terms, we evaluated our IDS using the accuracy value and the detection rate. Accuracy (ACC) is defined in the following equation as the number of intrusions over the total number of events.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

On the other hand, the detection rate (DR) is the probability of an alarm given of all the actual intrusions.

$$DR = \frac{TP}{TP + FN}$$

According to the aforementioned definitions, the results of our experiments are summarized in Table I. The experimental results indicate that our IDS can detect anomalies of the Android system with relative accuracy and detection rate to 85,55% and 81,56% respectively.

TABLE I. EXPERIMENTAL RESULTS OF IDS

Total NetFlows	Malicious NetFlows	ACC	DR
260456	45631	84,31%	81,06%
247104	31192	87,06%	86,82%
223455	27382	82,93%	80,49%
146009	21429	85,32%	83,44%
133674	17151	82,51%	80,93%
103444	16260	84,28%	81,17%
80554	13236	83,56%	80,99%
65932	11123	83,12%	80,54%
55828	9517	82,95%	80,51%
48235	6362	86,81%	80,52%
42493	5443	87,19%	80,61%
38033	4757	87,49%	81,02%
34447	4225	87,73%	80,04%

Finally, numerical results are given regarding the battery consumption as a collateral effect of an important metric that can evince the possibilities of our IDS operation in real time environment. The Sony Xperia P smartphone is equipped the Li-Ion 1305 mAh battery and the watt-hours of this battery are: 1305 mAh x 3.72 V / 1000 = 4.854 Wh. On the other hand, the Huawei Ascend P8 Lite smartphone is supplied the Li-Ion 2200 mAh battery and the watt-hours of this type battery are 2200 mAh x 3,8V / 1000 = 8,36 Wh. Fig 3 shows the energy consumption in Wh of the two devices during the IDS operation. According to the GSam Battery Monitor application for ten hours our IDS consumes 0,548 Wh and 0,9326 Wh of the energy of Sony Xperia and Huawei Ascend P8 Lite smartphones respectively.

V. CONCLUSIONS

Mobile device is emerged as a primary platform equipped with powerful sensing, computing, and networking capabilities. The popularity and the advanced functionalities of the modern mobile devices attract the attention of hackers and cyber criminals.

In this paper, we presented a lightweight IDS which is enhanced with a powerful MLP neural network for detecting malicious behaviors of the Android mobile devices. The experimental results indicate that our IDS can detect an anomaly in the Android operating system effectively. Specifically, the accuracy and the detection rate of the proposed system reaches 85,02% and 81,39% respectively. Future work aims at further improving the accuracy and the detection rate of the IDS, by taking into account more critical traffic features such as user's touch patterns and behaviors.

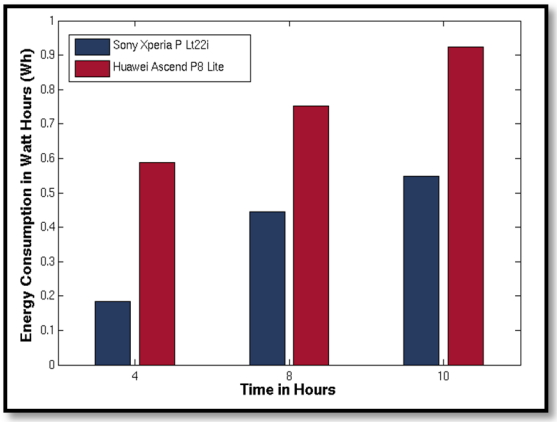

Figure 3: Energy consumption of smartphones Sony Xperia P Lt22i and Huawei Ascend P8 Lite.

REFERENCES

[1] H. Saïdi and A. Gehani, "Smartphone security limitations", Proceedings of the 2011 Workshop on Governance of Technology, Information, and Policies - GTIP '11, 2011.

[2] E. Maxwell, "Open Standards, Open Source, and Open Innovation: Harnessing the Benefits of Openness", Innovations: Technology, Governance, Globalization, vol. 1, no. 3, pp. 119-176, 2006.

[3] Hua and K. Sakurai, "A SMS-Based Mobile Botnet Using Flooding Algorithm", Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication, pp. 264-279, 2011.

[4] A. Schmidt, F. Peters, F. Lamour, C. Scheel, S. Çamtepe and Ş. Albayrak, "Monitoring Smartphones for Anomaly Detection", Mobile Networks and Applications, vol. 14, no. 1, pp. 92-106, 2008.

[5] F. Yuan, L. Zhai, Y. Cao and L. Guo, "Research of Intrusion Detection System on Android", 2013 IEEE Ninth World Congress on Services, 2013.

[6] M. Ghorbanian, B. Shanmugam, G. Narayansamy and N. Idris, "Signature-based hybrid Intrusion detection system (HIDS) for android devices", 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC), 2013.

[7] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, "'Andromaly': a behavioral malware detection framework for android devices", Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161-190, 2011.

[8] T. Draelos, D. Duggan, M. Collins and D. Wunsch, "Adaptive critic designs for host-based intrusion detection", Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290).

[9] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection", IEEE Communications Surveys & Tutorials, vol. 12, no. 3, pp. 343-356, 2010.

[10] Y. Yu, Y. Ge and G. Fu-xiang, "A neural network approach for misuse and anomaly intrusion detection", Wuhan University Journal of Natural Sciences, vol. 10, no. 1, pp. 115-118, 2005.

[11] A. Hoglund, K. Hatonen and A. Sorvari, "A computer host-based user anomaly detection system using the self-organizing map", Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, 2000.

[12] S. García, M. Grill, J. Stiborek and A. Zunino, "An empirical comparison of botnet detection methods", Computers & Security, vol. 45, pp. 100-123, 2014.