International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 01 | Jan 2020 www.irjet.net p-ISSN: 2395-0072

Local Security Enhancement and Intrusion Prevention in Android Devices

Santhosh Voruganti[1], Mohd Muawiz Siddiqui [2], Jallawaram Abhishek[3], Karnati Ramyakrishna[4]

[1]Asst.Prof, IT Department CBIT Hyderabad
[2]Student, IT Department CBIT Hyderabad
[3]Student, IT Department CBIT Hyderabad
[4]Student, Osmania University Hyderabad

Abstract - Android smart phones amount to 82% of the devices in the smartphone market. Every year there are more than 500 million new handsets sold, thereby granting Android the monopoly in the smartphone market. With an exponential increase in the number of users the risks associated with the phones also increases exponentially. In this paper, we use earlier approaches of host-based intrusion detection systems and behavior-based intrusion prevention systems for Android smartphones to design and implement a host-based, behavior-based intrusion prevention system, for Android smartphones. Our system uses net flow based clustering to identify anomalies and correlates further with the host-based features to verify malware intrusions in the Android system. Our goal is to provide versatile security for Android smartphones, offering detection of a wide range of attacks including denial of service attacks and probing. The system should be able to detect new attacks as well, thus providing scope for extending the method to other security solutions.

***Key Words*: Android; Host-Based; Behaviour-Based Information Security; Intrusion Prevention; Logit Boost.**

1. INTRODUCTION

The global telephony industry is witnessing an on- going proliferation of smartphones. A smartphone is an advanced mobile communication and a computing device which is shaping the way we communicate process and store information at work, at home and on the move. Smartphones are no longer mere voice communication devices. The considerable processing and storage capabilities are making their users to store and process private and business data. Data associated with these activities have significant value and over the recent years, smartphones are becoming an increasingly interesting target for cyber-criminals due to the wealth of personal data in them.

Over the past years, smartphone market share has increased rapidly and currently Android smartphones are dominating the smartphones. The popularity of Android open platform and the relative ease of programmability is making Android platform the lead malware target as well. However, this is possibly due to the unregulated third-party app stores for Android. Cybercriminals find the motivation to exploit smartphones as they store a wealth of personal data. Users tend to store more of their personal data on their smartphones, than on PCs; such as photos, videos, SMS, emails, and banking/shopping apps. Therefore, in this mobile computing era, protecting the safety of the smartphones is a top priority.

The advanced mobile communication devices such as smart phones are changing the way in which we communicate process and store data from any place. They evolved from simple mobile phones into sophisticated and yet compact mini computers. They are not just voice communication devices. Apart from browsing internet these devices can receive email send MMS messages, exchange information by connecting to other devices. They are also equipped with operating system, text editors, spreadsheet editors and database processors. As the capabilities of mobile devices evolve, their usage for processing and storing private and business data is likely to increase. Currently 200 million users are using smart phones worldwide and the number of people using smart phones will likely to increase to 1 billion in the next 2 years. That is approximately one sixth of the world population and equivalent to population of India.

As these devices can allow third party software's to run on them they are vulnerable to various threats like viruses, malware, worms and Trojan horses. Also a mobile device can initiate communication on anyone of its communication interfaces and also can connect to wide variety of wireless networks. Intrusion prevention mechanisms such as encryption, authentication alone cannot improve the security of the system. Already existing desktop based Intrusion Detection software's may not be good for mobile systems because of the memory consumption rate and power consumption. We need to come up with Intrusion detection systems that will not only improve the security of these systems but also reduce the processing overhead from the system.

In this paper, we aim to successfully overcome the shortcomings of existing systems i.e. Host-Based Intrusion Detection and Behavior Based intrusion Detection. A host-based system is faster and allows mobility of devices, thus making it a more feasible design trait for smart phones. A behavior based system is employed because it uses a learned pattern of normal network packets to identify active intrusion attempts and can adapt to new and original attacks, unlike knowledge-based systems. In behavior-based systems, feature reduction and selection reduces the number

of features given to the classifier, thus improving classifier accuracy. The application uses PCA for feature reduction, thus improving the classifier's accuracy considerably. Moreover, all the prevailing host- based systems detect device-dependent metrics and malware.

2. LITERATURE SURVEY

2.1 Host-based Intrusion Detection

A host-based IDS is capable of monitoring all or parts of the dynamic behavior and the state of a computer system, based on how it is configured. Besides such activities as dynamically inspecting network packets targeted at this specific host (optional component with most software solutions commercially available), a HIDS might detect which program accesses what resources and discover that, for example, a word- processor has suddenly and inexplicably started modifying the system password database. Similarly a HIDS might look at the state of a system, its stored information, whether in RAM, in the file system, log files or elsewhere; and check that the contents of these appear as expected, e.g. have not been changed by the intruders. Ideally a HIDS works in conjunction with a NIDS, such that a HIDS finds anything that slips past the NIDS. Commercially available software solutions often do correlate the findings from NIDS and HIDS in order to find out about whether a network intruder has been successful or not at the targeted host. Most successful intruders, on entering a target machine, immediately apply best-practice security techniques to secure the system which they have infiltrated, leaving only their own backdoor open, so that other intruders cannot take over their computers**.**

2.2 Behavior-based Intrusion Detection

Most security monitoring systems utilize a signature- based approach to detect threats. They generally monitor packets on the network and look for patterns in the packets which match their database of signatures representing pre-identified known security threats. Behavior analysis detection-based systems are particularly helpful in detecting security threat vectors in 2 instances where signature-based systems cannot (i) new zero- day attacks (ii) when the threat traffic is encrypted such as the command and control channel for certain Botnets.

A Behavior analysis detection program tracks critical network characteristics in real time and generates an alarm if a strange event or trend is detected that could indicate the presence of a threat. Large-scale examples of such characteristics include traffic volume, bandwidth use and protocol use.

Behavior analysis detection solutions can also monitor the behavior of individual network subscribers. In order for behavior analysis detection to be optimally effective a baseline of normal network or user behavior must be established over a period of time. Once certain parameters have been defined as normal, any departure from one or more of them is flagged as anomalous.

Behavior analysis detection should be used in addition to conventional firewalls and applications for the detection of malware. Some vendors have begun to recognize this fact by including Behavior analysis detection programs as integral parts of their network security packages.

2.3 Summary

The advantages and limitations of the studied systems lead to the following observations. A host-based system is faster and allows mobility of devices, thus making it a more feasible design trait for smart phones. A behavior based system is employed because it uses a learned pattern of normal network packets to identify active intrusion attempts and can adapt to new and original attacks, unlike knowledge-based systems. In behavior-based systems, feature reduction and selection reduces the number of features given to the classifier, thus improving classifier accuracy. Our model improves the existing version of 'Protego: A passive Intrusion Detection System for Android Devices. We use the same architecture and the procedure as Protego but the implementation is improved. While Protego has an accuracy of 92% using AdaBoost o build the classifier, our model has an accuracy of 97.8845% using Logit Boost algorithm. Another drawback of Protego that our model overcomes is that it could only detect any intrusions while our model can even stop the intrusions as soon as they are detected.

3. MODEL OVERVIEW

The proposed system is a behavior-based, host-based, passive intrusion detection system. We first remove the unnecessary attributes from the NSL-KDD dataset in the preprocess stage. We then use Principal Component Analysis to reduce features of the dataset. The generation of the packet capture file allows the application of feature reduction using principal components analysis. The system architecture of our model is same as Protego and has been depicted in Fig- 1. It consists of 5 subparts, organized in three basic modules:

- Classifier Training
- Packet Capture and Analysis
- Packet Classification

3.1 Classifier Training

The idea of training a classifier stems from the fundamental concept of supervised learning. Supervised learning is the machine learning task of inferring a function from labelled data. A classifier is an algorithm which allows one to define categories of nodes. An integral part of training the classifier is a predefined training set, which in this case is a modified version of the NSL-KDD dataset (reduced to suit the needs of smartphones). By running the dataset through the classifier

to train it, you can then run that trained classifier on unknown nodes or records to determine which class that node belongs to.

3.1.1 Classifier algorithm:

As accuracy is of utmost importance in intrusion detection systems, we decided to use an ensemble approach for classification. This is a composite model, made up of multiple classifiers which vote, and returns a class label prediction based on the collection of votes. After evaluating multiple classifier models for their accuracy, we zeroed in on Logit Boost (adds a cost functional of logistic regression,) as the classification algorithm, as it gave 97.982% accuracy.

3.2 Packet Capture and Analysis

Feature reduction is used for reduction of dimensions of data having high dimensionality. It consists of feature selection and feature extraction. Feature selection is a process of selecting a subset of features and discarding the features which are redundant or have a comparatively low or no information gain. This results in data with reduced dimensionality and thus increases efficiency of machine learning algorithms. We chose principal component analysis (PCA) to reduce the dimensionality of the data, as it has been widely applied to datasets in various scientific domains. PCA is a linear dimensionality reduction technique which explains the variance covariance structure of a set of variables, through a few new variables, that are linear combinations of the original variables. The new variables are obtained from the eigenvalues and eigenvectors of the data covariance matrix. The data is first normalized and its standard deviation is calculated. In the next step, eigen analysis is performed on the create independent orthonormal eigenvalue, eigenvector pairs.

Finally, the sets of principal components sort by Eigen value in descending order. The eigenvalue is a relative measure of the variance of its corresponding eigenvectors. We implemented our system by using different sets of the most significant features generated by PCA. The accuracy was found to be consistent for 10 or more most significant features i.e. 97.9865%. Thus, in the final version of the static application, we have selected the 10 most significant features obtained as a result of PCA.

3.3 Packet Classification

Before the records are classified, the trained model of the classifier - which was saved while training - is loaded. The connection records are then fed to the classifier. Based on this trained model, each of the record is then classified into two categories viz. normal and anomaly.

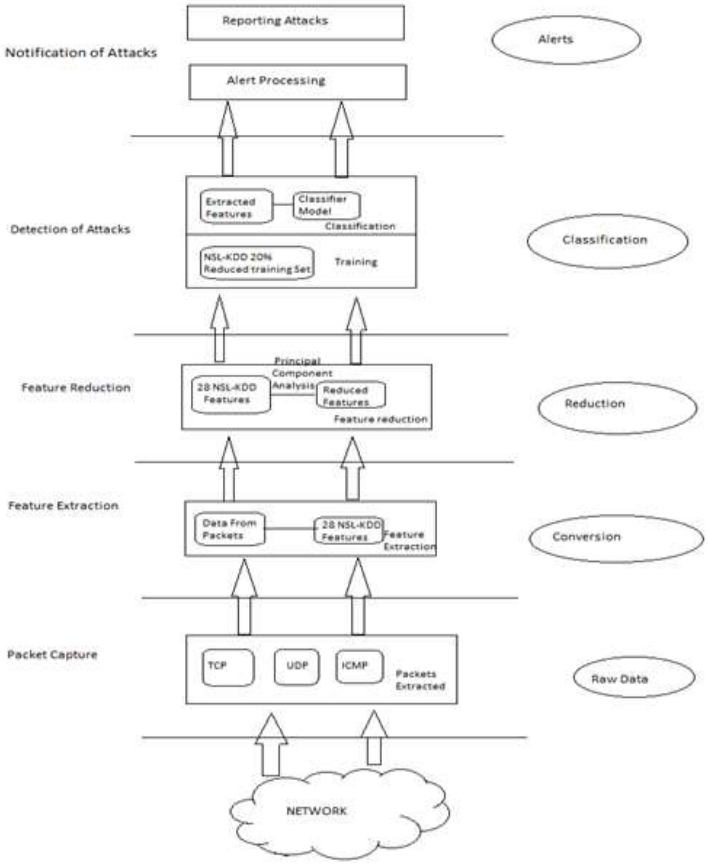

Fig-1: Architecture of the Model

4. IMPLEMENTATION

4.1 Hardware

The models for AdaBoost and LogitBoost are compared on WEKA, a data mining tool. The model is tested on an Android device running on Two Android smart phones running on Lollipop (5.0.2) the hardware specification of the phone is as follows:

- CPU- Dual-core 1.3 GHz (Cortex-A7)
- GPU- Mali 400
- RAM- 1GB

The hardware specification of the system to build and compare the models is as follows;

- CPU- Intel core I5
- GPU- NVIDIA GEFORCE 680
- RAM- 8GB

4.2 Software

Android Studio 1.1 was used for the development and testing of our model. Our model requires that the phone has a root access. We used tcpdump version3.9.8 to sniff packets.

Tcpdump is a command line tool which prints the description of contents of packets on a network interface. Busybox has been used for exploiting Unix tools. We use Weka, a machine learning tool used for data mining, for building and comparing the for the Android platform.

4.3 Dataset Description

In our classifier, we used the NSL-KDD data set, as it has solved some of the inherent problems of the KDDCUP'99 data set, which is a benchmark data set for evaluating intrusion detection systems. The training data set consists of 125,973 single connection vectors, each of which contain 41 features and is labeled as a normal or an attack vector. However, 13 features, like 'root_shell', 'su_attempted', 'num_file_creation', etc., are not related to smartphones and hence needed to be removed from the dataset to be relevant to our system. Removing these features manually could be done without affecting the training set as the features had low relevance and information gain.

feature name	description	type
hot	number of ''hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ''compromised'' conditions	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ''hot'' list; 0 otherwise	discrete

Table 2: Content features within a connection suggested by domain knowledge.

feature name	description	type
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have ''SYN'' errors	continuous
rerror_rate	% of connections that have ''REJ'' errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have ''SYN'' errors	continuous
srv_rerror_rate	% of connections that have ''REJ'' errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Table 3: Traffic features computed using a two-second time window.

Fig-2: Dataset Description

The above figure gives a brief description of the various attributes selected in the dataset. Out of the 41 attributes present in the NSL-KDD dataset, we remove 13 attributes which are not necessary in an android environment.

4.4 Model Creation

The Dataset is modified to suit the needs of an android device. After removing the attributes that are no longer required, Principal Component Analysis is done on the Dataset.

Once PCA is done on the Dataset we proceed to build the model for the classifier. We use Logit Boost algorithm to ensure maximum accuracy. The model is tested for varying number of iterations and the most accurate case is taken.

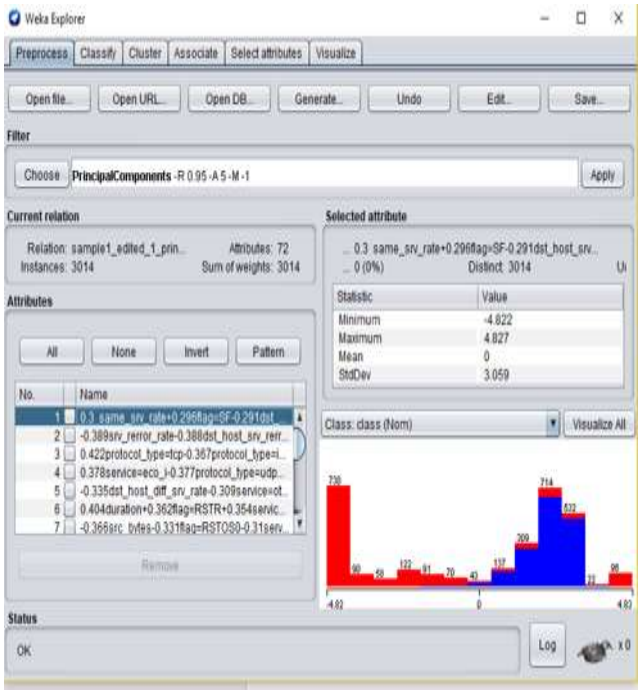

Fig -3: Principal Component Analysis done on Dataset

Sample paragraph Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

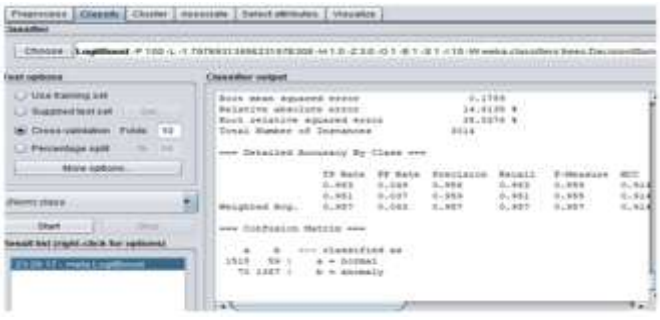

Fig-4: Working of the Algorithm to build the model

We compare the results between AdaBoost and Logit Boost for various iterations and the results are tabulated in the

form of a graph. Our assumption that Logit Boost is more accurate is proved to be true.

In the above figure, the Logit Boost algorithm is being used. We calculate Root Mean Square Error, Relative Absolute Error as well as Root Relative Squared Error. The detailed accuracy gives the True Positive, False Positive, Precision, Recall and other metrics to better understand the accuracy.

5. TESTING AND RESULTS

5.1 Evaluation

In order to evaluate the implementation of our model, we executed the application and checked the results.

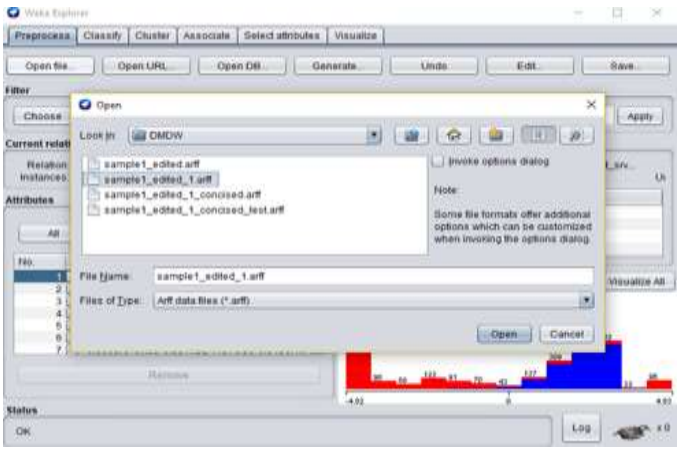

Fig-5: Loading the dataset

The above figure illustrates the loading screen and the options for loading the dataset into the WEKA tool.

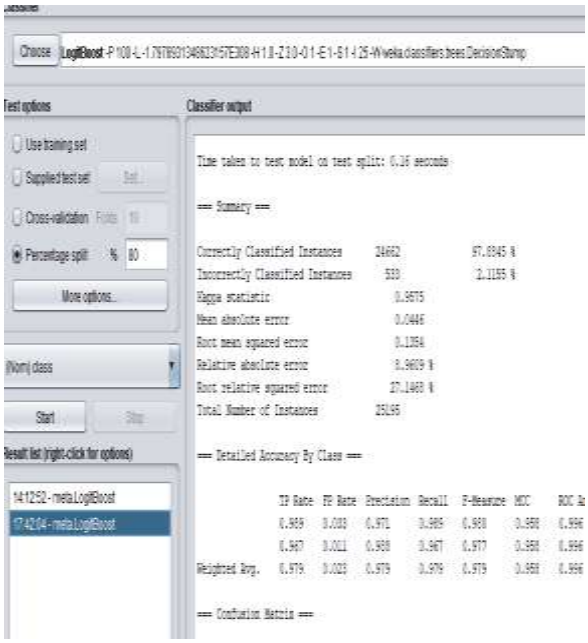

Fig-6: Accuracy for 25 iterations (97.8845%)

Fig-7: Performing LogitBoost in AndroidDevice

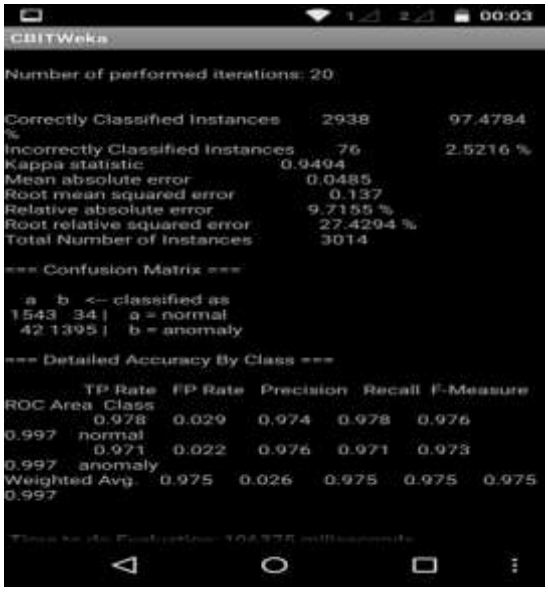

Fig-8: Building Model in Android device

In the above 2 figures, we redo the process of building the model for the classifier in the android device. We use an extension of WEKA for Android devices to facilitate this.

5.2 Results

The system correctly classifies the respective data with an accuracy of 97.8845%. The number of iterations and their corresponding accuracies are as follows:

- 10 features: 96.4199%
- 20 features: 97.3606%
- 25 features: 97.8845%

A graph comparing the accuracy of AdaBoost and Logit Boost is shown in Fig- 9.

5.3 Discussion

Although Protego is an efficient intrusion detection system, it fails to prevent those intrusions and our model overcomes this limitation.

Intrusion prevention: Our model can successfully prevent the preassigned set of intrusions trying to attack the Android Dataset. The Application gives a toast whenever any such intrusion is detected.

In the below graph we compare the accuracy between AdaBoost and LogitBoost algorithms. The X axis denotes the number of iterations and the Y axis gives the accuracy. From the graph we can conclude that with an increase in the number of iterations the accuracy of both AdaBoost and LogitBoost algorithms increase. But in comparison to AdaBoost algorithm (denoted by the green line), LogitBoost algorithm (denoted by the red line) is more accurate for every case which is studied.

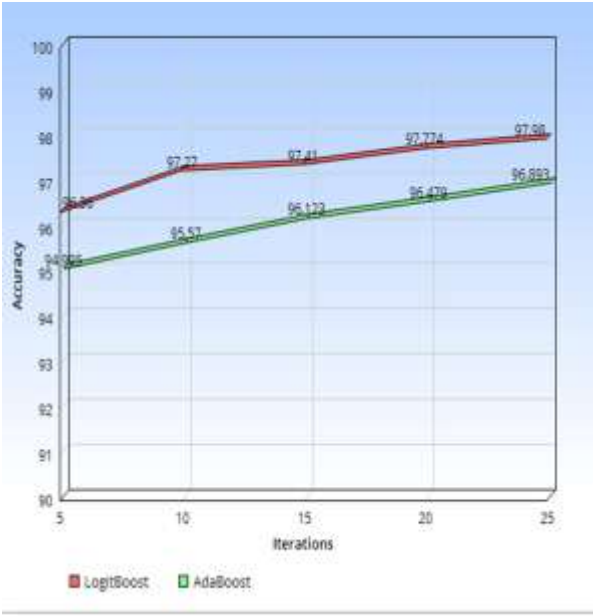

Fig-9: Comparison between LogitBoost and AdaBoost

6. CONCLUSIONS AND FUTURE SCOPE

In this work, we present a novel intrusion prevention method on Android smartphones. The system captures network traffic and classifies it to detect intrusions and unauthorized network activity on the host. The key focus of the study was to develop a system which could accurately detect any deviation from normal activity and cope with changing system behaviour, while imposing less overhead on the host system. The developed system is robustly able to detect denial of service attacks and probing attacks with an accuracy of 97.8845 %. The idea presented albeit not radical, more exploration is clearly required. We think that this system design would ameliorate the current smartphone security scenario. For future research directions, we believe that the system can be used in various environments, thus allowing a customized approach to security for smartphones.

The main limitation of this paper is that we need root access for running this app. The scope of this application can be extended so as to analyze real time data streams and torrent data packets. We can also include a provision to check the packets in a network connection in real time.

REFERENCES

[1]. Prachi Joshi "Protego: A Passive Intrusion Detection System for Android Smartphones", 2016 International Conference on Computing, Analytics and Security Trends (CAST), Dec 2016.

[2]. Dimitrios Damopoulos, "Intrusion Detection and Prevention Systems for Mobile Devices: Design and Development," Ph.D. Thesis, Dept. of Information and Communication Systems Engineering, University of the Aegean, Greece, 2013

[3]. Sherif, Joseph S., and Tommy G. Dearmond. "Intrusion detection: systems and models." In 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 115- 115. IEEE Computer Society, 2002.

[4]. Gupta, Kapil Kumar. "Robust and efficient intrusion detection systems.", Ph.D. Thesis, Department of Computer Science and Software Engineering, University of Melbourne, 2009.

[5]. Sanchez, Jaime. Building Android IDS on Network Level. DEFCON 21, 2013.

[6]. Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and Yael Weiss. "Andromaly: a behavioral malware detection framework for android devices". Journal of Intelligent Information Systems, pages 1–30, 2011. 10.1007/s10844-010-0148-x.

[7]. Dini, Gianluca, Fabio Martinelli, Andrea Saracino, and Daniele Sgandurra. "Madam: a multi-level anomaly detector for android malware." In Computer Network Security, pp. 240-253. Springer Berlin Heidelberg, 2012

[8]. Adigun, Abimbola Adebisi, Temitayo Matthew Fagbola, and Adekanmi Adegun. "SwarmDroid: Swarm Optimized Intrusion Detection System for the Android Mobile Enterprise." International Journal of Computer Science Issues (IJCSI) 11, no. 3 (2014).

[9].Elrawy, Mohamed Faisal, T. K. Abdelhamid, and A. M. Mohamed. "IDS in Telecommunication Network Using PCA." arXiv preprint arXiv: 1308.2779 (2013).

[10]. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten; The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. (2009).

[11].Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali-A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. (2009).

[12]. Kayacik, H. G□es, A. Nur Zincir -Heywood, and Malcolm I. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets." Proceedings of the third annual conference on privacy, security and trust. (2005).