International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 25 No. 2 March 2021, pp. 136-147

Optimization Of An Anomaly Based Intrusion Detection System On Smartphone

Ugwuanyi, Peace Nkiruka[1], Okafor, Loveth Ijeoma[2], Nelson Ogechukwu Madu[3], Agbo, Jonathan Chukwunwike[4], Barnnabas Bundepuun Orndiir[5], Doowuese Kate Faafaa[6], Ugwu Nnaemeka Virginus[7], Ani Chinonso Darlington[8], Anigbogu Kenechukwu Sylvanus[9]

[1]Centre for Distance and e-learning, University of Nigeria, Nsukka
[2,5,6,7]Department of Computer Science, University of Nigeria, Nsukka
[3]Department of Computer Science, University of Lagos.
[4]Department of Computer Science, University of Kairouan, Tunisia.
[8]Lacart Store Limited
[9]Nnamdi Azikiwe university, Awka , Anambra State
Computer Science department.

Abstract – Android OS is one of the widely used mobile operating systems. There is a huge increment in malware applications in android phones. This is an effort gear towards detecting malicious activities. This paper proposes a technique that can detect any illegal activities in smart phone using anomaly based. It analyzes system calls' logs and also the conduct of an app and afterward produces signatures for malware conduct. Intrusion detection system (IDS) is meant to be a software application which monitors system activities and detect any intrusion actions or operations. We proposed a system that will detect any illegal/malicious intrusions in Smart phone using anomaly based approach. Tshis approach is based with respect to viewing the conduct of the gadget by monitoring various parameters and the status of the segments of the gadget. This paper adopts the object oriented analysis and design method (OOADM). This models real world processes, operations and the data is also represented in a more flexibly, efficiently and realistically behaviour. Object-Oriented examination gives a simple progress to mainstream Object-Oriented programming dialects, for example, Java and C++. The proposed system will help to monitor users Android phone by detecting, authenticate intrusion and also log and mail alert of an attempt to the user's phone through the phone number and email.

Keywords – Anomaly based, Intrusion detection system, Smart phones, and malicious activities.

I. INTRODUCTION

Mobile gargets are quickly developing as famous apparatuses with progressively powerful computing, networking, and sensing capabilities. Smart device is the current overwhelming individualized computing gadget with such a significant number of features that are practically equal to smaller than normal PCs, for example, calls, short message administration, mixed media informing administration, email, video calling, voice correspondence, versatile banking, record trade, web perusing and so forth. As indicated by Pew Research Center in 2015, about 43% of the worldwide populace utilizes a cell phone gadget [1]. IDS is a sort of security programming intended to consequently caution heads when a person or thing is attempting to bargain data framework through

pernicious exercises or through security approach infringement. It additionally screens the Internet to scan for any of the most recent dangers which could result in attack. There are a number of IDS such as Dynamic, Static and Anomaly based approach among others. This project adopted the Anomaly based approach on Android. Anomaly based is a conduct based malware location system and it consistently screens the various highlights of the gadget state, for example, battery level, CPU use, arrange traffic, and so on. Estimations are taken during running and are then provided to a calculation that groups them likewise.

II. THEORETICAL BACKGROUND

The theoretical background consists of the breakdown of the technological approach used in the development of this work. The main components of Android gadget are: hardware, operating system, and application runtime. Androids keep running on various types of hardware arrangements. These incorporate Smart-phones, tablets, set-up boxes and so on. Android working framework is made out of the center which is expand over Linux Kernel. The Android application runtime is made in Java programming language and keep running in Dalvik virtual machine.

2.1 Android Application Development

Android applications are basically written in Java with assistance for their own neighborhood libraries written in C. The Java source code gets requested to a Dalvik Virtual Machine (DVM) executable byte code, which is taken care of in a DEX file. The DEX file, the Manifest, all benefits, the certificates and have libraries for the application are packaged to a ZIP record file with the APK suffix. This APK file is given through the Google Play to the customers. The source code of an Android application isn't available in clear substance while emptying an APK file.

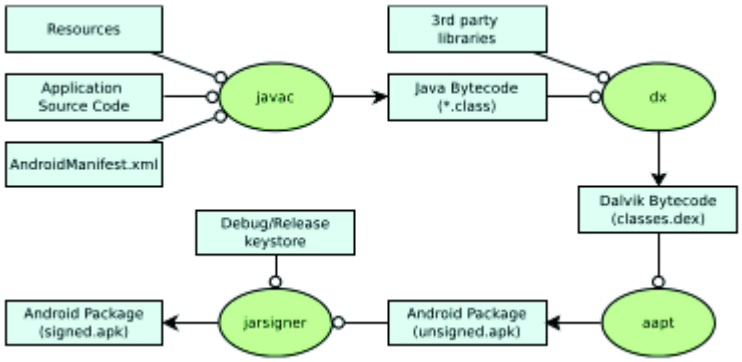

Figure 2.2: Build process of an Android application (Adapted from 2)

2.1.2 Android Malware Techniques

1. Static approach eliminates features from the application's archive without executing the application to recognize vindictive models. From the application's source code, various features are isolated, for instance, approvals, convey beneficiaries, APIs, purposes, data stream, control stream, gear fragments, etc. The most by and large used static features are the Permission and API calls. Since these are taken out from the application AndroidManifest.xml and effect the malware acknowledgment rate to a serious level.

2. Dynamic approach examines the lead of the application in a run time condition and screens the application's dynamic direct and system responses. It executes the dubious application inside a controlled circumstance often called sandbox. The unique highlights checked are sort out affiliations, work calls, resources use, system calls, etc.

3. Anomaly based approach is based concerning seeing the lead of the contraption by observing different boundaries and the situation with the sections of the device. Andromly is a lead based malware area framework and it reliably screens the different features of the contraption state, for instance, battery level, CPU use, orchestrate traffic, etc. Assessments are taken during running and are then given to a computation that bunches them in like manner. CrowDroid and AntiMalDroid are two interesting peculiarities based instruments used for malware revelation in Android devices. The first depends after separating structure calls' logs while the last analyzes the direct of an application and subsequently makes marks for malware lead. SMS Profiler and iDMA are two contraptions used to distinguish half-baked use of system benefits in iOS. [3]

2.1.3 Essential features of a good (anomaly based) IDS.

1.	Confidentiality	A great intrusion recognition framework guarantees that the information is not uncovered to unapproved people, procedures, or frameworks.
2.	Integrity	It guarantees that the information is protected with respect to its significance, fulfilment, consistency, planned use, and connection to its portrayal
3.	Availability	Ensuring that the information and framework are open and usable to approved people or potentially forms.
4.	Accountability	Ensuring that exchanges are recorded with the goal that events might be reproduced and followed to clients or procedures.

2.2 Review of Related Literature

We survey some of the existing approaches used by researchers for detecting intrusions. Various approaches have been used to detect intrusions and they can be roughly grouped into dynamic, static and anomaly based. Below, we give a brief review of research studies/ that have been conducted using several approaches.

[4] In this paper we have proposed another system to acquire and examine cell phone application movement. They found that observing framework calls is one of the most precise strategies for deciding the conduct of Android applications. The creator built up a lightweight customer called Crowdroid. This application utilizes publicly supporting way of thinking where a client sends non individual yet conduct related information of every application they use to the server. This is trailed by malware recognition dependent on the call vectors by the server. The exploratory outcomes completed by the writer had 100% location rate for self-composed malware.

According to [1], work was done for recognizing the malware as a standout amongst the most feared dangers to a developing PC and correspondence innovation. It recognized the: class of malware, malware characterization calculations, malwares exercises and methods for anticipating and expelling malware in the event that it inevitably taints framework. When a framework has been undermined by a malware, an aggressor would then be able to dispatch their assault through a few devices like bundle sniffer, port scanner, weakness scanner, password crackers among others.

In [2], reviewed calculations in information mining utilizing (learning disclosure in database) KDD-cup 99 (this is a sort of dataset use to assess the performance of an intrusion detection framework precedent least square-bolster vector machine (LSSVM – IDS) in the order of assaults and analyzed their outcomes which have been come to. A lot of calculation was assessed on KDD dataset and it was utilized for recognizing the classes of assaults. The AI calculations utilized as classifiers for KDD cup informational collection don't offer much guarantee for recognizing client to root (U2R) and remote to nearby (R2L) assaults that is the shortcoming.

According to K. Mani, et al [5] presented a review of different strategies and methods in understanding the ideas of filter based feature selection. The different sorts of: include choice, classes of administered feature selection and correlation of filter and wrapper. It limits excess, evacuates commotion and augments the significance of the objective in order. The multiplication of enormous information inside spaces presents extraordinary test to information mining.

In [6], proposed a mutual data based calculation that logically chooses the ideal element for grouping. The assessment results demonstrates that the feature selection calculation contributes progressively basic highlights for least square help vector machine based – interruption discovery framework to accomplish better precision and lower computational expense contrasted and the best in class strategies. The shortcoming is that "huge information" hinder the whole identification process and may prompt inadmissible grouping precision because of the computational challenges in taking care of such information.

According to [3], manages the importance of each component in KDD 99 intrusion recognition dataset to the discovery of each class. Their exact outcomes uncovered that a few highlights (hot Login, number of Compromised conditions, number of record creation tasks, visitor login) have no pertinence in interruption identification. Harsh set level of reliance and reliance proportion of each class were utilized to decide the most segregating features for each class.

According to [7], proposed a novel way to deal with break down factually the system traffic crude information. The enormous measure of crude information of real system traffic from the Intrusion Detection System is investigated to decide whether traffic is an ordinary or hurtful one. The issue is currently transformed into the sensor system to build the precision recognition rate, on the grounds that no hunt spaces are diminished.

Yousef Farhaoui, et al [8] present the various structures of IDS, measures that help to characterize the level of adequacy of IDS and the ongoing work of institutionalization and homogenization of IDS. The system enables us to update the analyzer to find conceivable new assaults or varieties of assaults. Their restrictions don't ensure 100% security, ridiculous and the disservice of this arrangement is the rate of false positives because of strange or unordinary conduct of clients, who are not really hurtful.

Heba Fathy, et al [9] proposed NID models utilize diverse keen calculations and feature selection and extraction methods and approve another component choice methodology "Bi-Layer social based element choice methodology", which relies upon the conduct of the order precision as indicated by positioned include. The test results on four proposed NID models, demonstrates the models points of interest of upgrading the identification exactness and testing speed by diminishing the component measurement space. One of the significant research difficulties for building elite NIDS is managing information containing huge number of features.

In [10], proposed an outfit model that is blend of C4.5 and Classification and Regression Tree (CART) as hearty classifier for arrangement of assaults. The proposed ensemble model gives tasteful precision as 99.67% and 99.53% if there should be an occurrence of double class and multiclass NSL-KDD informational index separately. A standout amongst the most significant efficiencies in the KDD informational collection is the colossal number of repetitive records, which makes the learning calculations be one-sided towards the incessant records, and in this way keep them from learning rare records which are normally increasingly destructive to systems, for example, U2R and R2L assaults.

According to [11], proposed a framework so as to improve the security of the portable applications which will assess the versatile applications security dependent on the distributed computing stage and information mining. The assessment results demonstrate that it is reasonable to utilize distributed computing stage and information mining to confirm all put away applications routinely to sift through malware applications from versatile application markets. The moving of the security usefulness into the cloud could likewise be dangerous, if not all pieces of the phone can be imitated into the cloud.

In [12], evaluated data in respect to classifiers configuration, utilized dataset, feature extraction, clustering strategies, exactness location measures and so on. The work of numerous and cross breed classifiers, improves the precision of the grouping and encourages understanding troublesome issues. The shortcoming is that binomial or typical (measurable circulations) can't delineate example acknowledgment conduct, which implies that customary systems of parametric techniques may not work.

According to [13], proposed another solid half breed technique for an oddity system based IDS (NIDS) utilizing artificial bee colony (ABC) and Adaptive Boosting calculations (ADA Boost) so as to pick up a high recognition rate with low false positive rate. The exactness and identification rate of this technique has been improved in correlation with unbelievable strategies. The shortcoming is the bogus alert report of interruption to the system and interruption identification precision that occurs because of the high volume of system information.

This [14] paper removes six kinds of data Permission, Intent channel, Intent channel, Process name, Intent channel, number of reclassified authorization from show documents and uses them to recognize Android malware. Results show that the technique can identify obscure malware tests that are imperceptible by a straightforward mark based methodology. This methodology is modest to execute in light of the fact that solitary the show document is examined.

In [15], Presented a quick, adaptable, and precise framework for Android malware recognition and family distinguishing proof dependent on lightweight static investigation. DroidSieve utilizes profound review of Android malware to construct compelling and strong highlights reasonable for computational learning. Their discoveries show that static investigation for Android can succeed in any event, when gone up against with obscurity systems, for example, reflection, encryption and progressively stacked local code. While major changes in qualities of malware stay a to a great extent open issue, DroidSieve stays flexible against best in class muddling systems which can be utilized to rapidly infer new and grammatically unique malware variations.

In [16], Presents a consent based Android malware identification framework, APK Auditor that utilizes static examination to describe and order Android applications as benevolent or vindictive. APK Auditor comprises of three segments: A mark database

to store extricated data about applications and examination results, an Android customer which is utilized by end users to give application investigation demands, and a focal server liable for speaking with both mark database and cell phone customer. 8762 applications were utilized to test framework execution. Result shows that APK Auditor can recognize most notable malwares and features the ones with a potential in around 88% precision with a 0.925 specificity.

This [17] study introduced a cell phone double guard insurance system that permits official and elective Android Markets to recognize vindictive applications among those new applications that are submitted for open discharge. This structure comprises of servers running on mists where designers who wish to discharge their new applications can transfer their product for confirmation reason. The confirmation server first uses framework call measurements to recognize potential malevolent applications. After confirmation, on the off chance that the product is perfect, the application will at that point be discharged to the significant markets. The exploratory outcomes utilizing 120 test applications (which comprise of 50 malware and 70 typical applications) show that we can accomplish 94.2% and 99.2% exactness with J.48 and Random woodland classifier individually utilizing this system.

2.2.1 Comparison of the articles reviewed showing strength/weakness.

S/n	**System/Article**	**Weakness**	**Strength**
1.	Malware Detection, Supportive Software Agents and Its Classification Schemes [1].	When a framework has been undermined by a malware, an aggressor would then be able to dispatch their assault through a few devices like parcel sniffer, port scanner, defencelessness scanner, password cracker among others.	It recognized the class of malware, malware arrangement calculations, malwares exercises and methods for forestalling and expelling malware in the event that it in the end contaminates framework.
2.	Building an intrusion detection system using a Filter-based feature selection algorithm [5].	These "huge information" hinder the whole recognition process and may prompt unacceptable arrangement exactness because of the computational challenges in dealing with such information.	The shared data based calculation can logically chooses the ideal component for grouping and it demonstrates that our element determination calculation contributes progressively basic features for LSSVM-IDS to accomplish better precision and lower computational expense contrasted and the best in class techniques.
3.	A Review on Filter Based Feature Selection [6].	The expansion of huge information inside numerous area presents phenomenal difficulties to information mining	It limits repetition, expels commotion and boosts significant to the objective arrangement. It is a pre-preparing stage which improves the precision, speed, information quality and comprehension. It likewise serves to lessen dimensionality and computational assets.
4.	Evaluation of Different Data Mining Algorithms with KDD CUP 99 Data Set [9].	The AI calculations utilized as classifiers for KDD cup informational index don't offer much guarantee for distinguishing client to root (U2R) and remote to neighborhood (R2L) assaults.	A lot of calculation was assessed on KDD dataset and it was utilized for identifying the classifications of assaults.

5.	Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features [3].	Observational outcomes uncovered that a few features (hot Login, number of Bargained conditions, number of record creation activities, visitor login) have no importance in interruption location.	It has high identification rate of AI calculation on them.
6.	A simple statistical analysis approach for Intrusion Detection System [7].	The issue is currently transformed into the sensor system to expand the exactness discovery rate, in light of the fact that no pursuit spaces are diminished.	The gigantic measure of crude information of genuine system traffic from the Intrusion Detection System is investigated to decide whether traffic is an ordinary or destructive one.
7.	Creating a Complete Model of an Intrusion Detection System effective on the LAN [8].	These frameworks are not ready to discover new intrusion or interruptions for which marks don't exist.	The framework is use for distinguishing unusual conduct or suspicious exercises to undermine the genuine task of the framework.
8.	Computational Intelligence in Intrusion Detection System [9].	Eite of NIDS posture real difficulties for these frameworks.	The exploratory outcomes on four proposed NID models, demonstrates the models favorable circumstances of upgrading the location exactness and testing speed by lessening the element measurement space.
9.	Intrusion Detection System for Classification of Attacks with Cross Validation [10].	individuals model are not proficient to recognize the bogus caution rate with better precision	The proposed gathering model gives acceptable exactness as 99.67% and 99.53% if there should be an occurrence of parallel class and multiclass NSL-KDD informational index individually.
10.	Malware Detection System for Android Mobile Applications [11].	The moving of the security usefulness into the cloud could likewise be risky, if not all pieces of the phone can be imitated into the cloud	The assessment results demonstrate that it is down to earth to utilize distributed computing stage and information mining to confirm all put away applications routinely to sift through malware applications from portable application markets.
11.	Evolution of Intrusion Detection Systems Based on Machine Learning Methods [12].	Binomial or typical (factual dispersions) can't delineate example acknowledgment conduct, which implies that conventional procedures of parametric methodologies may not work.	The work of numerous and cross breed classifiers, upgrades the exactness of the grouping and encourages understanding troublesome issues.
12.	Anomaly network-based intrusion detection system using	The bogus alert report of intrusion to the system and intrusion	The precision and detection rate of this technique has been

	a reliable hybrid artificial bee colony and AdaBoost algorithms [13].	discovery precision occurs because of the high volume of system information.	improved in examination with amazing strategies.

III. System Requirement Specification

The design tool used in this work is UML which help the developer to properly visualize, specify, construct and document all the components that make up the system, during the design stage. The UML tools used are: the use case diagram, class diagram, activity diagram and state transition diagram. Below are the diagrams:

3.1 Use Case Diagram

Use case charts are typically alluded to as conduct graphs used to portray a lot of activities (use cases) that some framework or frameworks (subject) ought to or can perform and furthermore demonstrates an angle of an outside client (actors). It accentuation more on what a framework does instead of how it is finished. Use-Cases are associated with situations. A situation is a case of what happens when somebody collaborates with the framework. It is demonstrating a Use-Case connection between android intrusion discovery frameworks (AIDS), User, Alert Agent [18].

The role of each actor represented in the use case diagram is described as:

1. Android intrusion detection system (AIDS): This is the proposed application; its role is to detect, Filter and authenticate intrusion.
2. User: He/she will download/install the app, login and also check log records for intrusion.
3. Alert Agent: The agent audit, log and mail alert of an attempt to the user.

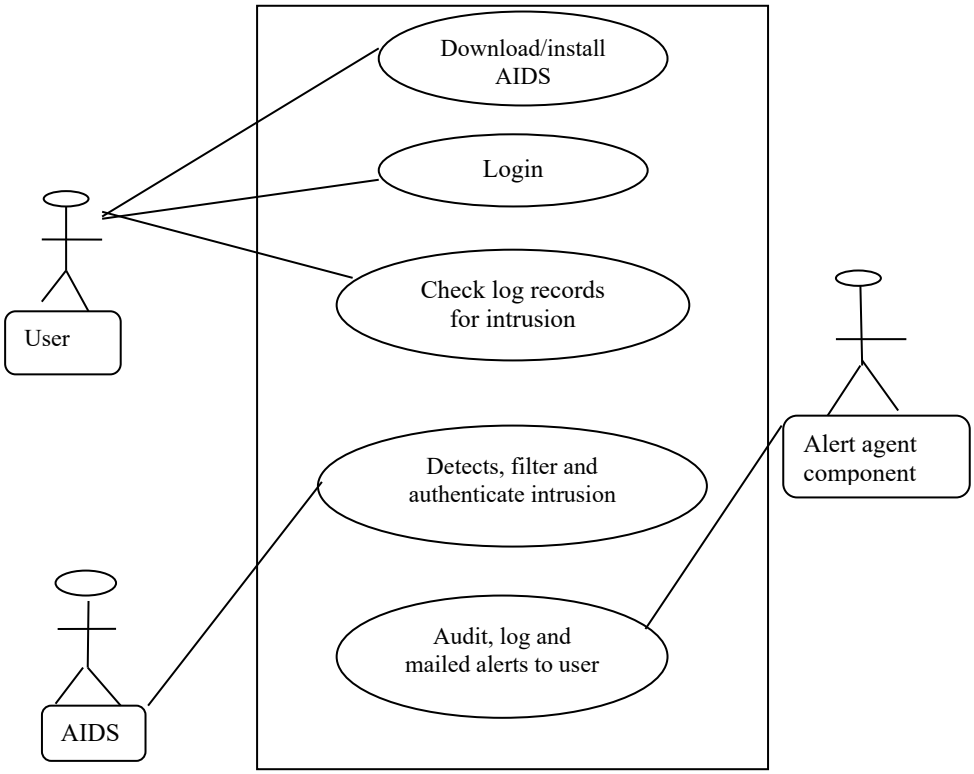

Figure 3.5 Use-Case diagrams of the system

3.2 Class Diagram

Class Diagram in is a sort of static structure outline that portrays the structure/review of a framework with its classes, their properties, activities (or techniques), and the connections among objects. The class chart is the principle building square of article situated displaying. It is utilized for general reasonable demonstrating of the deliberate of the application, and for nitty gritty displaying making an interpretation of the models into programming code.

The class diagram shown in fig 3 covers all components like the sampler, alert agent, database, and verifier and AIDS

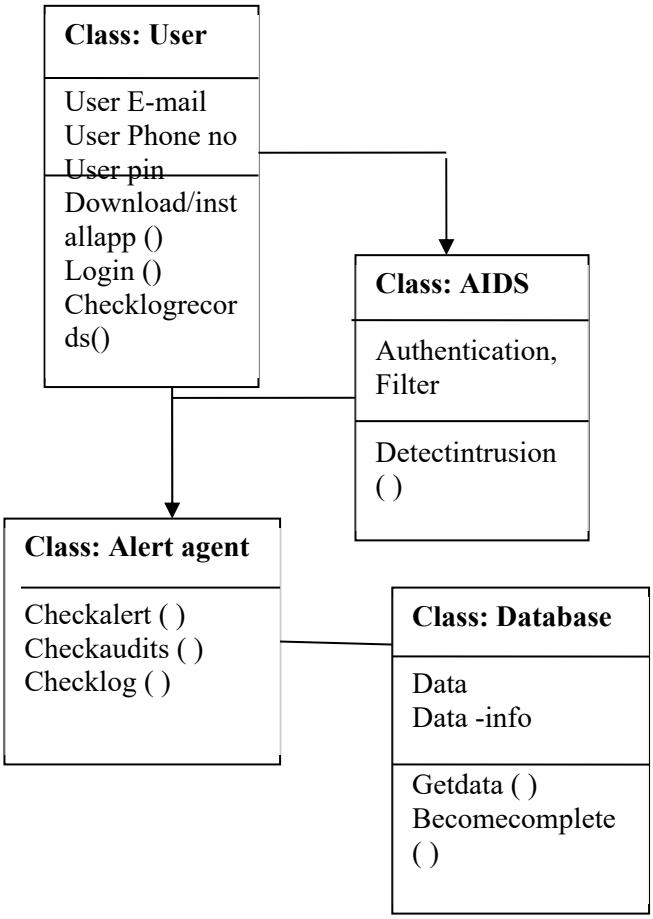

Figure 3.6: Class Diagram of the proposed system

3.3 Input Design

This refers to the design of the medium through which all the data that the system needs can be supplied to the system, with an explicit declaration of their data types. This design is basically “form design” which contains the users’ registration data that could be needed for the system to carryout specified operations or tasks. Figure 3.3 is the illustration of the input design of the user registration form.

Figure 3.3: Illustration of User Registration

3.4. Output design

The output design determines how the processed data is to be displayed. The design shows the alert SMS sent to the user in form of a string contain the day, month, time and year of the attempt. Figure 3.3 illustration of the Output Design of an alert SMS of an attempt.

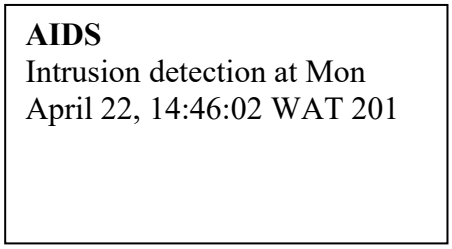

Figure 3.4: illustration of the Output Design of an alert.

IV. Choice of Development Environment

The integrated development environment (IDE) used in the development of this work is the

Android studio v2.2.0, on which the source codes are written, compiled and uploaded on Google Play Store. Android Studio offers numerous features that improves profitability when building Android applications, for example, Gradle-based system which is use to manage all dependencies (to build, test, run and package your app), Android Virtual Device (Emulator) also helps run and debug apps in the Android studio. The programming languages employed in this project are Java while Realm database management system was used.

4.1 Implementation Architecture

The implementation architecture of the AIDS is represented in figure 4.2 below. It is made up of the various components of the software modules and their linkages.

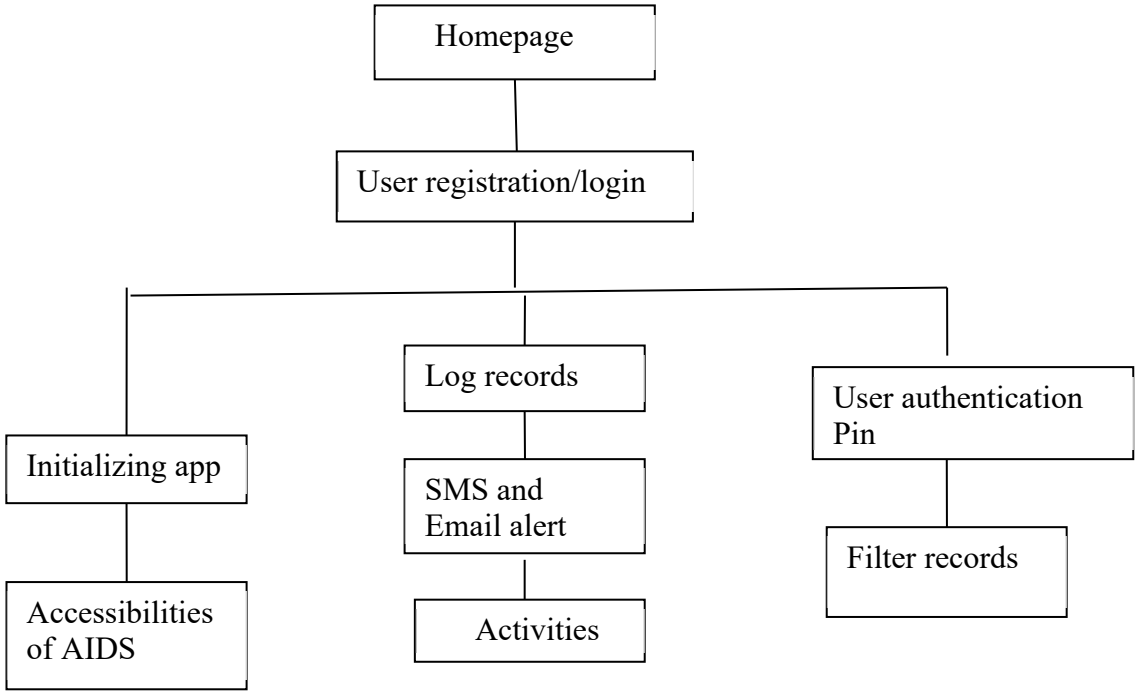

Figure 4.1: The Implementation Architecture.

4.2 Software Testing

Software testing is a process of executing a program application with the intention of finding the software bugs. This system has undergone series of test during the development stage in order to detect compilation errors and have them removed. The testing is of different phases:

4.2.1 Testing levels

The testing adopted is of different phases:

i. **Unit testing**: Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code. Testing done during the development phases in order to remove runtime errors or bugs, this error was corrected. This testing is important because it helped in testing source code while the application is being built by catching bugs early in the development cycle.

ii. **Sub system testing**: This was done by the developer, After coding each module, the module was tested, for instance after coding the SMS alert, the application was ran to check if the SMS alert is working before proceeding to code another module.

iii. **User Acceptance testing:** After coding and compiling AIDS as APK (Android application package) file. The app was tested by some users by downloading and installing AIDS on Android phone to know if it meets the requirements of detecting malicious activities before uploading it on Google Play Store for users.

They made few observations, they tested the app and realised that no message of intrusion send to their email. This error was corrected and the system is working efficiently.

iv. **Integrated testing:** System testing involves testing after the combination of various subsystems component to produce the required software, at this stage, the entire app was tested. After the registration process, the user logs in. If there is

any attempt of an illegal user trying to login, an SMS and Email alert will be sent to the user phone number and email address and also a log record of all attempt will be kept.

V. CONCLUSION

The accomplishment made in the advancement of this project could profit the immense Users of Android phones. Android IDS is introduced in order to identify intrusions when other defensive measures fall flat, by inactively observing system events and searching for security related issues. This research work gives a successful and productive procedure to identify noxious exercises (attempt of authentication, any access of files/app is kept, records of logs) in Android Phone. The proposed system, Android intrusion detection system (AIDS) is an application that keeps running on an Android OS. We propose an Android Intrusion identification system that recognizes noxious acts precisely and the application is utilized by Android phone users. AIDS was developed using Android Studio, Android SDK (software development kit) written with Java. The Object Oriented Analysis and Design Methodology (OOADM) were used for the analysis, design and development of the system and Unified Modeling Language (UML) to model the system.

REFERENCES

[1] Adebayo Olawale, Surajudeen, M.A.Mabayoje, Amit Mishra, Osho Oluwafemi, “Malware Detection, Supportive Software Agents and Its Classification Schemes”. International Journal of Network Security & Its Applications (IJNSA), Vol.4 no.6, pg 33, 2012

[2] Franklin Tchakounte, “A Malware Detection System for Android,” Universitat Bremen, 2015.

[3] Belal Amro, College of Information Technology, Hebron University, " malware detection techniques for mobile devices". International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol.7, No.4/5/6, December 2017

[4] I. Burguera and U. Zurutuza, “Crowdroid : Behavior-Based Malware Detection System for Android,” *Proc. 1st ACM Work. Secur. Priv. Smartphones Mob. devices (SPSM ’11). ACM, New York, NY*, pp. 15–26, 2011.

[5] Mohammed A, Ambusaidi, Xiangjian He, ,Priyadarsi Nanda, Zhiyuan Tan,” Building an intrusion detection system using a filter-based feature selection algorithm”. IEEE transactions on computers, vol 1, no 1 pg 1-3, 2014.

[6] K.Mani, P.Kalpama, “A review on filter based feature selection”. International Journey of Innovation Research in Computer and Communication Engineering, Vol 4, issue 5, pg 9147, 2016.

[7] A.A. Waskita, H. Suhartantoy, P.D. Persadhazy, L.T. Handoko, “A simple statistical analysis approach for Intrusion Detection System”. Center for Development of Nuclear Informatics-National Nuclear Energy Agency, pg 1, 2014.

[8] Yousef Farhaoui, Ahmed Asimi, “Creating a Complete Model of an Intrusion Detection System effective on the LAN”. International Journal of Advanced Computer Science and Applications *(IJACSA)*, Vol. 3, No. 5, pg 1-2, 2012.

[9] Heba Fathy, Ahmed Mohamed Eid, Prof. Afaf Abo, El Ftouh Saleh, Prof. Aboul-Ella Hassanien, “Computational Intelligence in Intrusion Detection System”. Faculty of Science, Al-Azhar University for Obtaining the Degree of Doctor of Philosophy in Computer. pg 147-150, 2013.

[10] Akhilesh Kumar Shrivas, Prabhat Kumar Mishra, “Intrusion Detection System for Classification of Attacks with Cross Validation”. International Journal of Engineering Science Invention, Vol 5, Issue 9, pg 5, 2016

[11] Mr. Akash J Wadate, Prof. N. R Chopde, Prof. D. R. Datar, “Malware Detection System for Android Mobile Applications“. International Journal of Engineering Research and General Science, Vol 4, Issue 1, pg 21-22, 2016.

[12] Abdulla Amin Aburomman, Mamun Bin Ibne Reaz, “Evolution of Intrusion Detection Systems Based on Machine Learning Methods”**.** Australian Journal of Basic and Applied Sciences, Vol 7, no 7, pg 46, 2013.

[13] Mehrnaz Mazini[a], BabakShirazi[b], IrajMahdavi[b]. “Anomaly network-based sintrusion detection system using a reliable hybrid artificial bee colony and Ada Boost algorithms”. Journal of King Saud University, pg 799-806, 2018.

[14] R. Sato, D. Chiba, and S. Goto, “Detecting Android Malware by Analyzing Manifest Files,” *Proc. Asia-Pacific Adv. Netw.*, vol. 36, pp. 23–31, 2013.

[15] G. Suarez-tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "DroidSieve : Fast and Accurate Classification of Obfuscated Android Malware," *ACM Conf. Comput. Commun. Secur.*, 2017.

[16] K. Abdullah, D. Ibrahim, and C. Aydin, "APK Auditor : Permission-based Android malware detection system," vol. 13, pp. 13–15, 2015.

[17] X. Su, M. Chuah, and G. Tan, "Smartphone Dual Defense Protection Framework : Detecting Malicious Applications in Android Markets," *Mob. Ad-hoc Sens. Networks (MSN), 2012 Eighth Int. Conf.*, pp. 153–160, 2012.