Available at: www.researchpublications.org

# Study of Malware Detection Technique for Apk and SDK File Using Artificial Immune

Mr. Bhushan P Kinholkar
*PG Student, Department of Computer Science Engineering*
*SSBT's College of Engineering and Technology, Jalgaon, India.*
Email:bhushank0029@gmail.com

Mrs. Nilima Patil
*Assistant Professor,Department of Computer Engineering, (M.E.-C.S.E)*
*SSBT'S College of Engineering and IT,Jalgaon, India.*
Email: np_1234@rediffmail.com

***Abstract:-*** **The word wide sharply increase in the number of smartphones user, the Android platform pose to becoming a market fugleman that makes the need for malware analysis on this platform an urgent issue. The current Artificial Immune Based malware detection systems they focus on traditional computers that uses information from OS or network, but the smartphone software behavior has its own structure and semantics. Current research cannot detect malware in smartphone exactly and efficiently. To detect these problems, in this paper only study of artificial immune algorithm, capitalize on earlier approaches for dynamic analysis of application behavior as a means for detecting malware in the smartphone as like sdk or android phone as like apk. An Artificial immune based smartphone and android phone malware detection Framework is brought forwards and a prototype system is implemented, that the system can obtain higher detection rate and decrease the false positive rate.**

***Index Terms-*** **Artificial immune, smartphone, malware, android**

## I. INTRODUCTION

Malware detection based on software running behavior and its status, and found that the acts of and attempts on the system. Current malware detection is mainly concentrated in the traditional desktop computer, made a lot of research, but relatively small for the new smartphone malware detection. Traditional malware detection system for the analysis of the detection information from the log of the system is running. New type of smartphone system running log obtain the more difficult compared to desktop class operating system and network logs, it has more complex structure, usually has a unique semantics. Data mining, neural networks, machine learning technology based desktop class operating system level malware detection model is usually not taken into account the characteristics of the smartphone, there are some disadvantages of the smartphone operating system malware detection for sdk and apk file.

The biological immune system has a good variety tolerance, and immune memory, distributed parallel processing, self organizing, and self learning. Adaptive and robustness, with the immune principle to build smartphone malware detection model can significantly improve the detection efficiency of the malware detection system of the smartphone and android phone. A smartphone malware detection technology based on artificial immune framework in this paper describe apk and sdk file protection. Classic artificial immune model based on the concept of the gene and antigen describe the combination of objects and the type of operation and operating software behavior; conditional maximum does not repeat principles used to extract the antigen and the use of unequal length string matching rules used for affinity calculation.

The field of Artificial Immune Systems (AIS) is concerned with abstracting the structure and function of the immune system to computational systems, and investigating the application of these systems towards solving computational problems from mathematics, engineering, and information technology. AIS is a sub-field of Biologically-inspired computing, and Natural computation, with interests in Machine Learning and belonging to the broader field of Artificial Intelligence.

Article is organized as follow sequence: Section 2 of the related work; Section 3 of the formal description of problem; Section 4 gives the detail description of key technology; Section 5 conclusion.

## II. RELATED WORK

In 2008 Behavioral Detection of Malware on Mobile Handsets [1] Mobile handsets, much like PCs, are becoming more intelligent and complex in functionality. They are increasingly used to access services, such as messaging, video music sharing, and e-commerce transactions that have been previously available on PCs only. However, with this new capability of handsets, there comes an increased risk and exposure to malicious programs (e.g., spyware, Trojans, mobile viruses and worms) that attempt to compromise data confidentiality, integrity and availability of services on handsets. In this approach, the runtime behavior of an application (e.g., file accesses, API calls) is monitored and compared against malicious and or normal behavior profiles. The malicious behavior profiles can be specified as global rules that apply to all applications, as well as fine-grained application-specific rules. Behavioral detection is more resilient to polymorphic worms and code obfuscation, because it assesses the effects of an application based on more than just specific payload signatures. For example, since encryption decryption does not alter the application behavior, multiple malware variants generated via run-time packers can be detected with a single behavior specification.

In 2009 Designing System-level Defenses against Cellphone Malware [2] Cellphones are increasingly becoming attractive targets of various malware, which not only cause privacy leakage, extra charges, and depletion of battery power, but also introduce malicious traffic into networks. In this work, seek system-level solutions to handle these security threats. Specifically, a mandatory access control based defense to blocking malware that launch attacks through creating new processes for execution. To combat more elaborated malware which redirect program flows of normal applications to execute malicious code within a legitimate security domain, using artificial intelligence (AI) techniques such as Graphic Turing test. Through extensive experiments based on both Symbian and Linux smartphones, that both system-level counter measures effectively detect and block cellphone malware with low false positives, and can be easily deployed on existing smartphone hardware. This mechanism is effective in defeating malware which execute malicious codes in new processes.Third, to combat automated malware which gain controls of existing processes to execute malicious codes, a more comprehensive defense which identifies and blocks malware using AI techniques such as Graphic Turing test(GTT). Using MMS as an example, through challenging auser, their approach differentiates whether a delivery of MMSmessage is legitimate (user-initiated) or illegal (malware initiated).

In 2010 [3] Taint Droid An Information Flow Tracking System for Real time Privacy Monitoring on Smartphones Taint Droid automatically labels (taints) data from privacy-sensitive sources and transitively applies labels as sensitive data propagates through program variables, files, and inter process messages. When tainted data are transmitted over the network, or otherwise leave the system, TaintDroid logs the data's labels, the application responsible for transmitting the data, and the data's destination. Such real time feedback gives users and security services greater insight into what mobile applications are doing, and can potentially identify be having applications. Frame work that allows users to monitor how third-party smartphone applications handle their private data in realtime. Many smartphone applications are closed source; therefore, static source code analysis is infeasible. Even if source code is available, runtime events and configuration continually dictate information use; real time monitoring accounts for these environment specific dependencies. to monitor privacy sensitive information on smartphones. Sensitive information is first identified at a taint source, where a taint marking indicating the information type is assigned. Dynamic taint analysis tracks how labeled data impacts other data in a way that might leak the original sensitive information. This tracking is often performed at the instruction level.

In 2010 [4] A Behavior based Malware Detection System for Cellphone Devices Computing environments on cellphones, especially smartphones, are becoming more open and general purpose, thus they also become attractive targets of malware. Cellphone malware not only causes privacy leakage, extra charges, and depletion of battery power, but also generates malicious traffic and drains down mobile network and service capacity. In this work they devise a novel behavior based malware detection system named BMDS, which adopts a probabilistic approach through correlating user inputs with system calls to detect anomalous activities in cellphones. BMDS observes unique behaviors of the mobile phone applications and the operating users on input and output constrained devices, and leverages a Hidden Markov Model (HMM) to learn application and user behaviors from two major aspects process state transitions and user operational patterns. Built on these, BMDS identifies behavioral differences between malware and human users. Through extensive experiments on major smartphone platforms, they show that BMDS can be easily deployed to existing smartphone hardware and it achieves high detection accuracy and low false positive rates in protecting major applications in smartphones.

In 2011[5] Andromaly a behavioral malware detection framework for android devices Andromaly framework for detecting malware on Android mobile devices. The proposed framework realizes a Host-based Malware Detection System that continuously monitors various features and events obtained from the mobile device and then applies Machine Learning anomaly detectors to classify the collected data as normal (benign) or abnormal (malicious). Since no malicious applications are yet available for Android, they developed four malicious applications, and evaluated Anomaly's ability to detect new malware based on samples of known malware. They evaluated several combinations of anomaly detection algorithms, feature selection method and the number of top features in order to find the combination that yields the best performance detecting malware on mobile devices in general and on Android.

## III RELIMINARIES

### *A. Formal Description*

Before carrying out further research on the smartphone malware detection based on artificial immune, the formal description, clear concepts and definitions are gave as follows.

*Definition* 1: Process Session (abbreviated as PS). After pretreatment of a smartphone process running procedure is defined as a process session, which contains the main information of access sequence of system API at a specific time period.

*Definition* 2: According to the concept of process session, self-defined as the normal process session, nonself defined as abnormal process session. Software usually uses the system API to access OS providing functions. The short sequence of system API access is used to determine the process session is self or nonself.

*Definition* 3: Gene Pool (abbreviated as GP). Define the type of operation (abbreviated as OP) and manipulate objects (abbreviated as OO) combination (OP OO) gene, and in accordance with the OP changes in the smartphone is divided into "read operations (Read critical resource)", "write operation (Create, Update critical resource), the use of read, write, denoted by R and W.

*Definition* 4: antigen and antibody [6] (abbreviated as Ag & Ab). Genetic composition of the antigen and antibody by the gene pool are defined as antigen and antibody.

Where type represents the type of detector, 1 for immature detector, 2 for mature detector 3 for memory detector, 4 for pending detector in the detection phase, it is a special kind of immature detectors; ab said antibody, namely the operation of short sequences; life represents the life of the detector, the different types of detectors have different life; count is used to define the match number of short sequences occur in the detector antibody collection; N is the set of natural numbers.

The dashed box in Figure 1 said the similarity of the two detectors, the division of two phases only for immature detectors and pending detector has nothing to do with the content outside the dashed box.

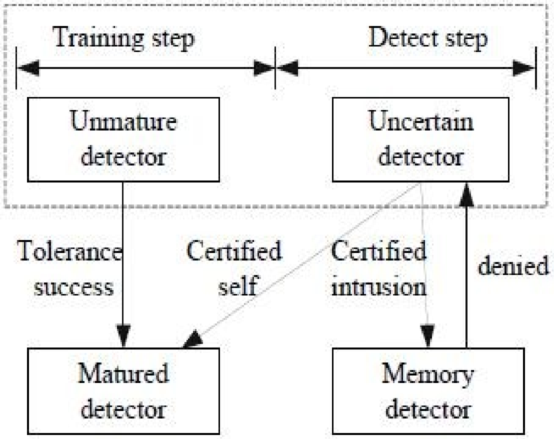

Figure 1: Transition relation diagram of four detector types

According to the definition given above, the smartphone malware detection based on artificial immune described as follows, under conditions of limited resources, the malware detection model detects any element in the collected process session set U of the smartphone, classify them into normal process session set NS or abnormal session set AS. False positive and false negative are two error types, the detection rate, false positives and false negative rate are used to assess the efficacy of the model in the detection process.

### *B Conditional Maximum not Repeat Principal*

Maximum does not repeat principle refers to the existence of an orderly repeat string S, and all the characters belong to S, contains the character table of the limited character of the sigma. Interception sub-string Sub in turn from the first element of S, the requirements of Sub is that the Sub is the longest one that not contains repeat characters and the length of it is not 1. However, this division of the string method may lead to a long string of up to repeat the string into its own. If a smartphone process session not repeated operations on the system API, the same problem will appear. Some constraints needed to be added up to the principle of limited to allow the longest substring of K, split strings that their length K equals or larger than K, even if all the characters in the string are not the same be divided by principle. This limit avoid detector antibody unlimited growing. The process of using conditional maximum does not repeat principle to divide string is shown in Figure 2.

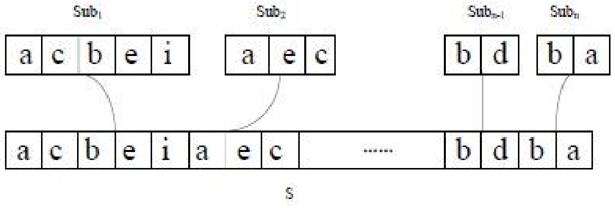

Figure 2: Maximum condition semantic diagram

## IV Framework Composition and Key Technologies

The detection framework include five main modules, they are log preprocessing module, the module of the gene pool, immature pending detector module tolerance, mature detector module and memory detector detection module. The model introduces the mechanism of the life cycle of the detector, can effectively control the size and model of the detector with good dynamic characteristics. The Model is shown in Figure 3.

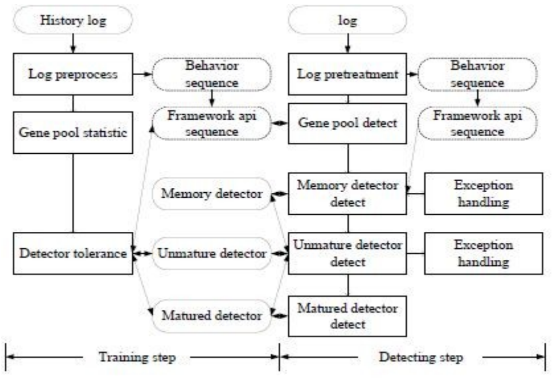

Figure 3: Architecture of Detection Framework Diagram

The square in Figure 3 [6], said processes and the dotted line ovals said temporary data collections, the solid line ovals said data collections, empty arrows indicate the flow of control, and black arrows indicate data flow. Detection model include the training and detecting phases. The training phase of the mission provide mature detector set for anomaly detection, the detection phase, according to the model in advance to learn the knowledge gained to implement malware detection task. Each stage is divided into two steps, and malware detection stage, for example: First, the system API audit log preprocessing, smartphone process session collections can be obtained, this process is the basis of the follow up operation; and then the anomaly detection, the use of the training phase detector set to detect abnormal behaviors based on user operation sequences.

The core of the anomaly detection is management of update, tolerance and death of the three types of detector antibody. The procedure is shown in Figure 4. The model consists of two processes one is the antigen processing process; the other is detector antibody evolution process. Co stimulatory in the Figure 4 [6] is get an affirmative answer information from user or system manager, the negative response or no response within a certain period of time known as the co stimulate fail.

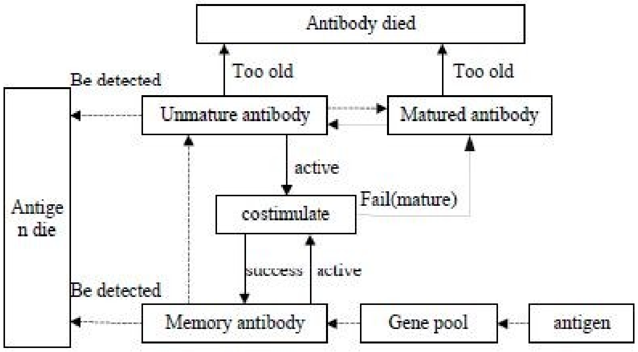

Figure 4: Detector antibody life circle diagram

*C. Antibody evolution process*

Immature and mature antibodies generated in the training phase of the collection are the basis and premise of the framework runs, there are three different types of antibody evolution process.

**Antibody memory**: When the memory antibody matches the antigen, the system determines that there is a malware alert waiting for system manager to confirm.

**Immature antibody:** Affirmative selection algorithm is used in the detection framework and immature detectors are the key to the discovery of malware. The immature antibody evolution, there are two flows to First, the lifetime did not match enough antibodies cannot be activated, too old to be deleted; in the life cycle is successfully activated, enter the co-stimulatory process, if co stimulatory successfully converted to memory antibody, to maintain the characteristics of malicious behavior, or converted to the mature antibody, to maintain the characteristics of legitimate behavior.

**Mature antibody:** Mature antibody is credible autologous collection, retention and maintenance of the collection is to maintain a modest scale of the collection of autologous antigen match the new immature antibodies. Mature antibody will be deleted because there is no match in the life cycle of a sufficient number of antigens. In order to ensure the diversity of the antibody, to avoid too little immature antibodies and mature antibody cannot detect non-self antigens, while the need to restrict the size of the antibody of the framework, the total number of antibodies if limited as a constant, mature antibody will be Scheduled scanning, and some of the poor quality of the antibody will be deleted.

*D. Gene Pool Management*

In this model, Management module of the GP is different from the traditional artificial immune immunity [6]. In actual situation, the operation of smartphone applications access the system API are often unavailable, so only collected software behavior log contains information extraction type of operation and manipulate objects. Therefore, the GP of the model is incremental, so when the model runs in reliable smartphone, once the discovery of new genes can be perceived abnormal behavior. The workflow of GP management module is shown in Figure 5.

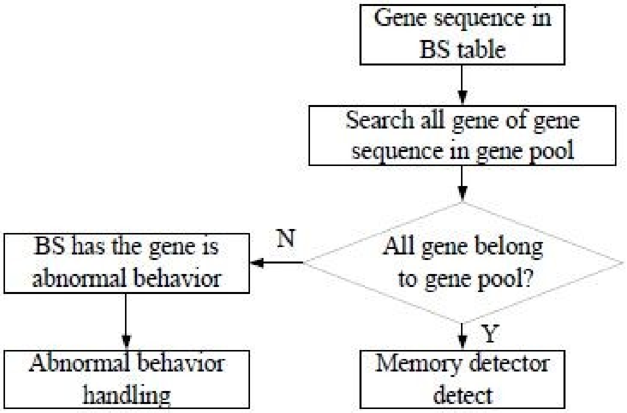
Gene sequence in BS table
Search all gene of gene sequence in gene pool
All gene belong to gene pool?
N
BS has the gene is abnormal behavior
Y
Abnormal behavior handling
Memory detector detect

Figure 5: Workflow of GP management module

*E. Immature Detector Detect*

While immature detector [6] matches the number of matching threshold , the detector will be activated and give an alarm signal, waiting for the judgment of the security administrator, if the co stimulatory, immature detector will convert the memory detector for rapid detection of malware otherwise, immature detector is converted to the mature detector, become a relatively stable collection of autologous. In order to limit the size of detector set, and also the immature detector, if the detector within the prescribed time does not match a sufficient number of antigen, the detector will be deleted. Immature detector module performs the detection process shown in Figure 6.

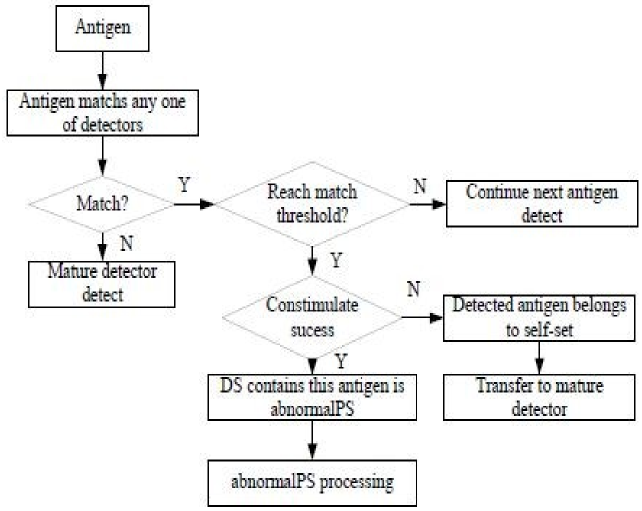
Antigen
Antigen matchs any one of detectors
Match?
Y
N
Mature detector detect
Reach match threshold?
N
Continue next antigen detect
Y
Constimulate sucess
N
Detected antigen belongs to self-set
Y
DS contains this antigen is abnormalPS
Transfer to mature detector
abnormalPS processing

Figure 6: Immature detector workflow diagram

## V CONCLUSION

A new framework to obtain and analyze smartphone application and their file activity in Android framework. In collaboration with the Android user community, it will be capable of distinguishing between benign and malicious applications of the same name and version, detecting anomalous behavior of known applications. that monitoring software behavioral activity in Android framework is a feasible way for detecting malware. According to the brief survey in section 2, they have seen that there're many different approaches to detect malware in traditional PC and malware in smartphone, such as Microsoft Windows Mobile, Nokia symbian, Apple ios and Google Android. That framework level Android sdk function call monitoring techniques can contribute to a deeper analysis of the malware, providing more useful information about malware behavior and more accurate results. On the other hand, more monitoring capability will place a higher demand on the amount of resources consumed in the device. They will implement our detection framework in as many different versions of smartphones as possible. Smartphones running our customized android operating system that embedded the above detection framework will have the opportunity to see their own smartphone behavior. an artificial immunology algorithm to distinguish between benign applications and their correspondent malware version. The results have been encouraging, although we need to address some open issues. First, the system would always separate the software process sessions in two sets even if there is no malware on it. The software process session mapping would change drastically whenever a malicious process session enters into the normal dataset. These issues require some manual check or further automatic analysis. This technique protect SDK and APK file from harmaly malware.

## REFERENCES

[1] Abhijit Bose, Xin Hu, Kang G. Shin and Taejoon Park, Behavioral Detection of Malware on Mobile Handsets, MobiSys 08 Proceedings of the 6th international conference on Mobile systems, applications, and services, pp. 225-238, June 2008.

[2] L. Xie, X. Zhang, A. Chaugule, T. Jaeger, and S. Zhu, Designing System-level Defenses against Cellphone Malware, In Proc. of 28th IEEE International Symposium on Reliable Distributed Systems (SRDS), pp. 83-90, 2009.

[3] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel et al, “Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones”, In Proceedings of the 9th USENIX conference on Operating systems design and implementation, OSDI'10, Berkeley, CA, USA. USENIX Association, pp. 1-6, 2010.

[4] L.Xie, X.Zhang, J. Seifert and Sencun Zhu, “pBMDS: A Behavior-based Malware Detection System for Cellphone Devices”, WiSec’10, Hoboken, New Jersey, USA, pp. 37 48, Mar 2010.

[5] 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, 10.1007/s10844-010-0148-x, pp. 1-30, 2011.

[6] M. Zhao, T. Zhang, J. Wang, and Z. Yuan, \A smartphone malware detection framework based on artificial immunology." JNW, vol. 8, no. 2, pp. 469 476, 2013.