International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

HOOK MALWARE DETECTION IN APK USING ADVANCED MACHINE LEARNING ALGORITHM

Badanapally Praneeth[1], Alla Naveen[2], Aluri Siddartha Reddy[3], Donthi Sainadh[4], Dr.V.Ajitha[5] [1,2,3,4]

B.Tech. Student, Department of Computer Science and Engineering,

Associate Professor, Department of Computer Science and Engineering,

Nalla Malla Reddy Engineering College, Hyderabad, India

***Abstract*—**As the use of Android OS is open source which is developed by Google and it is highly available to every person around the world. In parallel, cyber attacks are also initially growing day by day.Nowadays detection of malware in APK has become difficult.To overcome this we proposed a system to detect malware in APK by using Advanced Machine Learning Algorithms. Here we will be performing static and dynamic analysis techniques using
Python.The primary objective of this project is to Detect Hook
Malware.This malware has recently been attacked on many android users.We will be using Genetic algorithm, Super Vector Machine (SVM) and Artificial Neural Network.In association we will be using technique of Reverse engineering technique to derive features of two APK sets where one APK is good ware and the other one is malware .By extracting those features of two APK sets we will be able to analyze the malware and helps to detect malware.

Keywords—Android,Operating System(OS),APK,Malware,Hook Malware,Genetic Algorithm, SVM Algorithm,Reverse Engineering,Features,Static Analysis,Dynamic Analysis.

1. INTRODUCTION

Nowadays we are primarily using Android OS in our smartphones which was developed by Google in the year of 2003 and initiated the implementation in the year of 2005 in support of ease of access and user-friendly platform to interact with Android based mobile phones such as HTC Dream, Samsung ,Motorola,Google and many other giant smartphone companies started to integrate with Android OS.There is a lot of usage of technology in our daily lives.Based on particular agenda cyber attackers like Black hat hackers are getting expertize in creation of different malware attacks on Android users to steal all sensitive information like media ,bank details and personal details.A huge amount of data is being illegally sold in black market on dark web.Malware is a malicious software is a program code to steal all the necessary information unknowingly without any attention to android users.Malware has been categorized in many ways like ***Backdoor****:Code that allows the execution of unwanted ,potentially harmful,remote-controlled operations on a device.****Billing Fraud:*** *Code that automatically changes the user in an intentionally deceptive way.Mobile billing*

*fraud is divided into SMS fraud ,call fraud and Toll fraud.**Stalkerware (Commercial Spyware)**:Code that collects and/or transmits personal or sensitive user data from a device without adequate notice or consent and doesn't display a persistent notification that this is happening.**Denial of Service(DoS)**:Code that ,without the knowledge of the user, executes a denial-of-service(DoS) attacker or is a part of a distributed DoS attack against other systems and resources.**Ransomware:** Code that partial or extensive control of a device or data on a device and demands that the user make a payment or perform an action to release control..**Spyware:**Code that transmits personal data off the device without adequate consent like contact list,media,call log,SMS log and web history.**Trojan:** Code that appears to be benign,but that performs undesirable actions against the user.*

In August 2010,the first wild Android malware was reported by Denis Maslennikov,an employee of Kaspersky.Disguised in a **Windows Media Player Application**,***FakePlayer*** was sending SMS messages at the numbers 3353 and 3354,with each message costing about $5.In recent android malware attack the threat actor behind the BlackRock and ERMAC

Android banking trojans has unleashed yet another malware for rent called **Hook** that introduces new capabilities to access files stored on device using Virtual Network Computing (VNC) to create a remote interactive session.Now we are going to propose a system to handle and detect **Hook Malware** which is latest android malware attack.

2.LITERATURE REVIEW

Discovery of Hook Malware

ThreatFabric cybersecurity researchers have discovered a new type of Android malware known as 'Hook.' Hackers can use the malware to gain remote control of an infected device and steal sensitive information such as login credentials and financial information.

Hook, according to the researchers, is distributed via malicious apps downloaded from third-party app stores. When malware is installed on a device, it uses VNC to establish a real-time connection with a remote server (virtual network computing). This gives the hacker access to the infected device.

The malware is capable of a wide range of malicious actions, including audio and video recording, screenshot capture, and data collection about the device and its user. It also has the ability to intercept and redirect incoming and outgoing calls and messages, which could be used to steal sensitive information or commit financial fraud.

Hook can also circumvent two-factor authentication by intercepting and redirecting text messages, allowing hackers to gain access to online accounts.

Hook malware for Android works by intercepting and modifying the behavior of specific functions in the Android operating system. This enables malware to access sensitive information, such as login credentials or personal data, and perform actions without the user's knowledge or consent.

The malware accomplishes this by injecting code into system libraries or by gaining access to the device via the Android Debug Bridge (ADB). Once the malware has gained access to the device, it can steal sensitive information and transmit it to the attacker using various techniques such as keylogging or screen scraping. Furthermore, the malware can use the device to engage in other malicious activities, such as sending spam or joining a botnet.

"With this feature, Hook joins the ranks of malware families that are able to perform full DTO, and complete a full fraud chain, from PII exfiltration to transaction, with all the intermediate steps, without the need of additional channels," warns ThreatFabric.

3.METHODOLOGY

Mechanism of Malware

Firstly we need to understand how malware actually works.The workflow of a malware is different when we compare this from other malware..

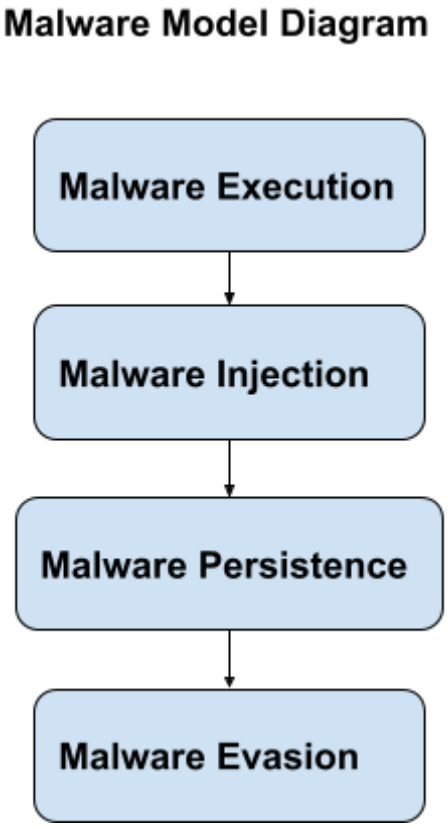

Fig.1:Workflow of Malware

The malware starts with the execution stage, where it is launched on the victim's system. It then proceeds to the injection stage, where it tries to insert itself into the victim's system.the persistence stage, where it tries to ensure that it stays active on the system, even after the victim reboots the system or runs a virus scan.

The malware achieves persistence by creating new processes, modifying system settings or registry keys, or installing itself as a system service.Finally, the malware enters the evasion stage, where it tries to avoid detection by antivirus software and other security measures. It does this by using various techniques such as encrypting its code, obfuscating its behavior, or mimicking legitimate system processes.

Mechanism of Hook

There are typically four stages of Hook malware attack as following

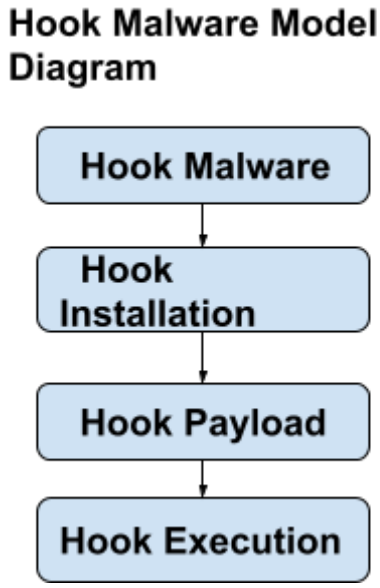

Fig.2:Workflow of Hook Malware

The hook malware starts with the installation stage, where it is installed on the victim's system. It then proceeds to the execution stage, where it tries to execute its payload.

The payload stage is where the hook malware inserts its malicious code into legitimate system processes or applications. This is achieved by hooking various functions and APIs used by the system or applications, for example, by intercepting network traffic or modifying system calls.

Once the hook malware has successfully hooked into the system or applications, it can perform various malicious activities, such as stealing sensitive data, logging user keystrokes, or downloading and executing additional malware.Hook malware mainly gets connected by using Virtual Network Computing (VNC).It remotely operates the device of a victim by a hacker.

Hook Malware Detection Using Genetic Algorithm, SVM and ANN

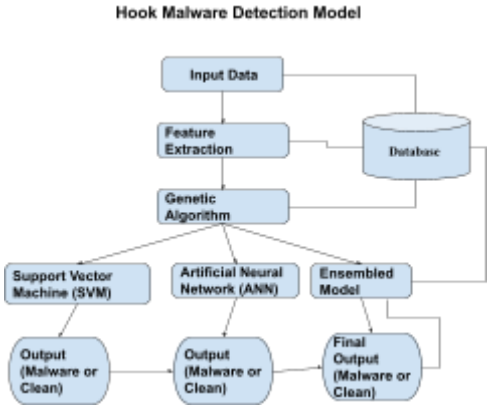

Fig.3:Workflow of Hook Malware Detection Using Genetic Algorithm,SVM and ANN

This diagram shows the different stages of the hook malware detection model. The input data is fed into the feature extraction stage, where relevant features are extracted from the data. The extracted features are then fed into the genetic algorithm stage, which optimizes the feature selection and parameter tuning process.The output of the genetic algorithm stage is then used to train two different models: an SVM model and an ANN model. Both models use the optimized features and parameters from the genetic algorithm stage to classify the data as either malware or clean.The outputs from both models are then combined using an ensembling technique to produce the final output. This final output represents the classification of the input data as either malware or clean. So we can use the dataset which is publicly available as follows:Malware Data Science (MDS) by Joshua Saxe and Hillary Sanders: MDS is a publicly available dataset of 1.2 million malware samples, including hook malware. The dataset also includes extracted features that can be used to train machine learning models for malware detection.Malware Corpus from the National Institute of Standards and Technology (NIST): NIST maintains a collection of malware samples that can be used for research purposes. The dataset includes a variety of malware families, including hook malware.Microsoft Malware Classification Challenge (MCC): MCC is a publicly available dataset of over half a million malware samples. The dataset was created for a competition to develop machine learning models for malware classification, and it includes hook malware.Kaggle Datasets: Kaggle hosts a variety of datasets, including several datasets on malware. Some of these datasets include hook malware samples and can be used to train machine learning models.Here features are obtained from two

APK using Androguard or APKTool

3. RESULTS

The results for the proposed system is obtained from the developed Machine Learning Model in the form of Binary Classification output.It gives an output as to whether an APK is a malware or not.

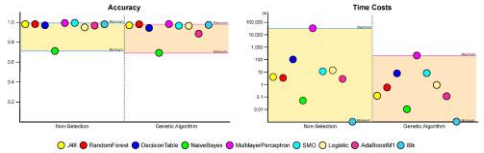

Fig.4: A comparison of machine learning performances between non-selection and genetic algorithm-based feature selection.

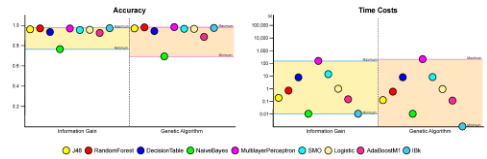

Fig.5: A comparison of machine learning performances between information gain-based and genetic algorithm-based feature selection.

Fig.6: Uploading training and testing dataset for detection of malware

Fig.7: Options for selecting the operation to be performed on given dataset which has Malware Detection and Performance Metrics

Fig.8: A result of performance metrics

In summary, we can conclude that using data without feature selection is the best choice in terms of the evaluation metrics. However, considering the time required, it is better to proceed with feature selection through genetic algorithms, with which we can expect better results than applying information gain. These results can be visualized as *Fig.:4* and *Fig.:5* .Finally we can expect the result in the form of *Fig:6,Fig:7* and *Fig:8.*

4. DISCUSSION

The forensic analysis that we have performed and whose results were outlined in the previous section has yielded a number of insights for the research and practice of malware detection. In this section, we summarize these insights and discuss how this empirical study could be instrumented in our work on malware detection.

A. Summary of findings

On Antivirus software: Our large-scale analysis of hundreds of thousands of Android applications with over 40 anti virus products have revealed that most malware are not simultaneously identified by several anti virus. Only a small subset of common malware is detected by most anti virus software. This finding actually supports the idea that there is a need to invest in alternative tools for malware detection such as machine-learning based approaches which are promising to flag more malware variants. *On malware business:* We have presented empirical evidence that malware was mass produced. This raises a number of questions leading to hypotheses on how malware developers manage to remain productive. The first hypothesis would be that malware is not written from scratch, thus providing an opportunity to detect malware by discovering the piece of code that was grafted to existing, potentially popular, apps.

B. Insights

Building a naive anti virus software: Exploring the rate of shared certificates within malware, we were able to devise a naive malware detection mechanism based on the appearance of a tagged certificate. In its simplest form, the scheme consists in tagging any application as malicious if the signing key has been already observed for a confirmed malicious application.

To assess this naive approach we have considered that in a first phase we have manually discovered all malware packaged before 01/Jan/2013 in our dataset. We consider for this step only malware that are detected by at least half of the anti virus products. Then based on the certificates recorded for the found malware, we arbitrarily tag as malicious all applications packaged after 01/Jan/2013 and that are signed with any of the flagged certificates. Below provides the results for this experiment. We were able to build a malware detector with a Precision of 84% (2, 166 false positives out of 2, 166 + 11, 460 tagged). While we succeed in flagging almost 1 actual malware out of 10, we only wrongly tag as malicious about 1 benign app in 100.

PERFORMANCE OF A NAIVE ANTI VIRUS SOFTWARE BASED ON CERTIFICATES

	Benign apps tagged	**Malware tagged**
Number	2.166	11460
Percentage	1.19%	8.82%

At the minimum, the obtained results show that our naive approach could be used by anti virus vendors to improve their recall, by being suspicious of more apps, and improve

precision by trusting apps signed with certificates that have been used in a large number of benign apps.

Localizing malware: Our findings on the potential mass production of malware could be leveraged in an approach of malware localization. Indeed, simultaneous development and packaging of malware suggests a redundant insertion of malware code in all applications. Thus, a similarity measure of the bytecode could allow to isolate this code and then locate it in other malware samples

5. Conclusion

In this paper, we implement an Android malware detection system based on
SVM,Genetic Algorithm, different from the traditional detection method, it can detect unknown Android applications based on machine learning. We extract various features with the method of static analysis and dynamic analysis using ANN. A new feature selection algorithm is also proposed to dispose of the raw features and our experimental result shows that the new method performs better with higher detection rate and lower error detection rate compared with the traditional detection approaches such as the detection method based on feature extraction.

.

6.ACKNOWLEDGEMENTS

Hook malware detection in apk using advanced machine learning algorithms will be having a lot of use cases in the near future. It is estimated that 90% of the search fields will be enabled with this system. It helps to test hook malware detection in an APK and we can protect the devices from getting compromised.

References

[1] **K. Allix, T. F. Bissyande´, Q. Jerome, J. Klein, R. State, and Y. Le Traon, "Large-scale machine learning-based malware detection: Confronting the "10-fold cross validation scheme" with reality," in *CODASPY '14*, 2014.**

[2] **S. Arzt, S. Rasthofer, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," in *Conference on Programming Language Design and Implementation (PLDI)*, 2014.**

[3] **A. Bartel, J. Klein, M. Monperrus, K. Allix, and Y. Le Traon, "Improving privacy on android smartphones through in-vivo bytecode instrumentation," Technical Report, May 2012.**

[4] **A. Bartel, J. Klein, M. Monperrus, and Y. Le Traon, "Dexpler: Converting Android Dalvik Bytecode to Jimple for Static Analysis with Soot," in *ACM Sigplan Workshop on the State Of The Art in Java Program Analysis (SOAP)*, 2012.**

[5] **J. Bickford, R. O'Hare, A. Baliga, V. Ganapathy, and L. Iftode, "Rootkits on smart phones: attacks, implications and opportunities," in *HotMobile '10*, Maryland, 2010.**

[6] **J. Brodkin, "On its 5th birthday, 5 things we love about android," Nov. 2012, http://arstechnica.com/gadgets/2012/11/ onandroids- 5th-birthday- 5- things- we- loveabout- android/.**

[7] **S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "Xmandroid: A new android evolution to mitigate privilege escalation attacks," Technische Universita¨t Darmstadt, Technical Report TR-2011-04, Apr. 2011.**

[8] **I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android," in *SPSM '11*, Chicago, Illinois, USA, 2011, pp. 15–26.**

[9] **P. P. Chan, L. C. Hui, and S. M. Yiu, "Droidchecker: analyzing android applications for capability leak," in *WISEC '12*. Tucson, Arizona, USA: ACM, 2012, pp. 125–136.**

[10] **L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, "Privilege escalation attacks on android," in *ISC'10*. Boca Raton, FL, USA:Springer-Verlag, 2011, pp. 346–360.**

[11] **A. Desnos, "Android: Static analysis using similarity distance," in *HICSS '12*. Washington, DC, USA: IEEE Computer Society, 2012, pp.5394–5403.Pittsburgh, PA: Springer-Verlag, 2011, pp.**

[12] **Hook Malware Discovery by BLEEPINGCOMPUTER**

[13] **Categories of malware by Google**

[14] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," in Proceedings 2014 Network and Distributed System Security Symposium, 2014