BEHAVIORAL MODEL FOR LIVE DETECTION OF APPS BASED ATTACK

LAKSHMI MUPPALLA[1]

CHANDU DELHI POLICE[2]

[1]PG scholar, [2]Associate professor, Department of Computer Science Engineering
Priyadarshini Institute of Technology & Science, Tenali, Guntur

ABSTRACT: Smartphones that are equipped with application platforms are experiencing significant growth and recognition. Numerous security hazards have arisen as a result of the extensive utilisation of various applications. Permission control attacks, phishing attacks, spyware attacks, botnets, malware attacks, and privacy leakage attacks are among the threats. Additionally, other vulnerabilities include the compromise of data confidentiality, invalid access control, and invalid authorization of applications. Throughout this paper, an application-based approach to attack modelling and detection is proposed. As a result of On the basis of the smartphone's application execution, a novel attack vulnerability is identified. The attack modelling process entails the use of a vulnerable application by an end-user to initiate an attack. Hidden from the end-user's view, the vulnerable application is deployed in the background of the smartphone. Consequently, the confidential information is accessed. The assault model is addressed by the proposed technique of an Application-based Behavioural Model Analysis (ABMA) scheme in the detection model. In order to execute the intrusion detection procedure, the model implements application-based comparative parameter analysis. The parameters of power, battery level, and data utilisation are employed to estimate the ABMA. The analysis is conducted using three distinct configurations: mobile data, WiFi, and a combination of the two, as determined by the source internet accessibility. The efficacy of the proposed model is confirmed and illustrated by the simulation results.

Keywords: smartphone ABMA, power, battery level

I. Introduction

In recent years smart phone application models have explosively increased from personnel to professional applications including education, online shopping, net banking, and healthcare. The platform of these applications has massively increased the threat of attacks by compromising trustworthiness and security capabilities [1]-[3]. Third party application marketing is one of the major threat, wherein interested application can be installed by the end-user. However, the applications from these platforms can prove menaces with the advent

of vulnerable breaches. Various attacks were identified that can prove detrimental and have adverse effects on the overall security of the information concerned to the smart phone. The jamming attack is one of the prime issues against time-critical applications. The attack exposes the in transit confidential information to the intruders [4]. Inaudible voice attack manipulates voice controllable device with unnoticeable characteristics while operating modulation technique using ultrasonic carriers [5]. The camera based attack proves a serious security threat to the multimedia applications of smart phones [6]. The side-channel attack exploits the leakage data to limit the data confidentiality on smart phones [7],[8]. Pin inference attack is identified as the privacy threat for the devices controlled by smart phones [9]. Indirect eavesdropping attack is another possible menace that makes use of acoustic sensing to execute the attack on the smart phone [10]. Permission control is one of the primary countermeasures against the possible security risks in smart phones.

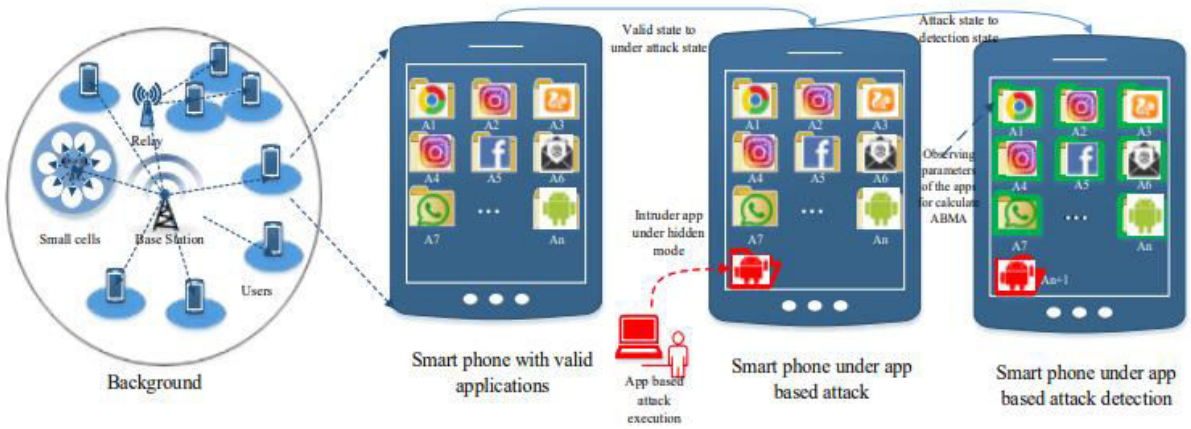

Figure 1: System model of application-based attack modeling and detection

The permission control enhances the security by incorporating conditional restrictions on the particular executions performed by the applications. Various permission control methodologies were formulated including context sensitive permission control [11], user driven access control [12], permission control using crowd sourcing [13], and Sig PID (Significant Permission Identification) [14]. However, the major limitation associated with the permission control technique is that the targeted functionality of the application is restricted such that the desirable and undesirable private data transmission is not well differentiated.

II. Literature Review

The current security improvement techniques in view of smartphone application platforms are briefly summarized in Table I. In [18], earlier version applications have been discovered as the source of vulnerable threats of an attack. To counteract the attack possibility, Driodskynet has been developed as a tool to find out and evaluate the applications with security risks from the application installation source such as playstore. In [19], the possible security menaces are located in the android operating system having inter-component communication. The component-level data flow analysis technique has been executed to recognize the caller and the callee on the basis of the data dependencies. However, the communication based attacks are identified by the parameter of the intent abnormality. In [20], a self-defending mechanism has been formulated to allow the repackaged applications to manifest automatically. The scheme encrypts the portion of the application code during the compile-time and the ciphertext code is decrypted at the run time. In [21], an antiphishing scheme MobiFish has been proposed for smartphone platforms. The strategy involves the validity verification of applications, webpages, and other persistent accounts. The validation is obtained by comparing the claimed identity with the actual identity. In [22], end to end caller ID verification technique has been devised by evaluating the current smartphone network infrastructure. A CallerDec application has been designed as an ID spoofing detection tool for android based smartphones to evaluate validation and effectiveness of the mechanism.

III. Proposed Model

The attack modeling defines the execution of the attack by making use of the possible application-based vulnerability. The intruder makes use of the vulnerable application to initialize the attack execution. The intruder provides the installation link on the other installed applications of the cellular smartphone in the form of an advertisement or the pop up option similar to apps based phishing attacks. For any response of the user, the vulnerable application starts to download in the background. The already installed application is assumed to be linked to the play store. In other words, during the internet accessibility and the processing of the already installed applications, the vulnerable application breaches via other downloaded installed application targets the access in the database of the cell phone.

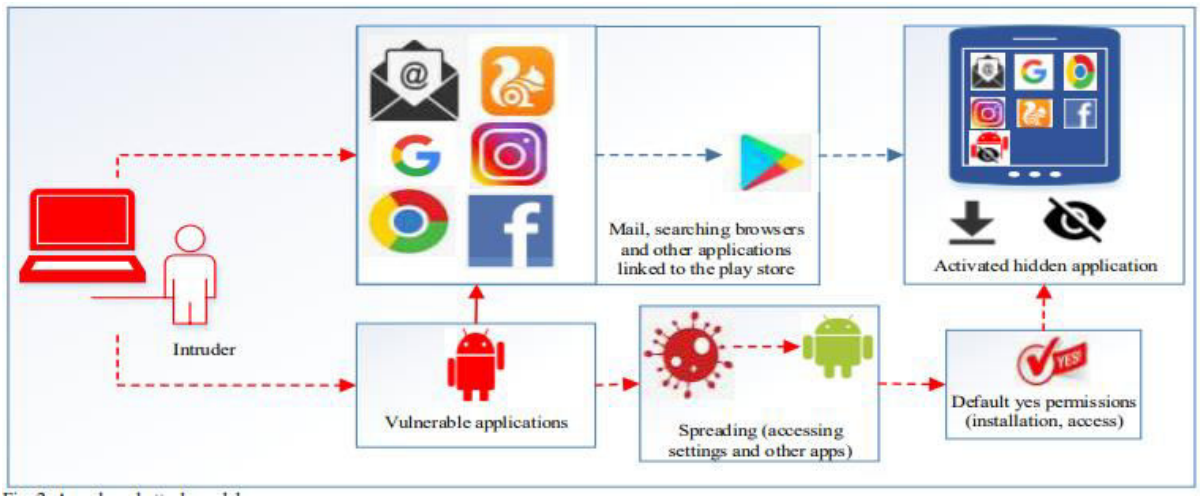

Figure 2: Apps based attack model

The vulnerable application has the ability to perform the function of spreading. Spreading is defined as, accessing settings and other applications. The vulnerable application is incorporated by the default 'Yes' accessibility to make the procedure devoid of the permissions such as permission control vulnerabilities. The whole process results in the installation of the intruder application under hidden visibility. The hidden visibility is defined as the disability mode where the intruder application is not visible on the user account and the list of applications on the smartphone. The mechanism of attack modeling is shown in Fig. 2.

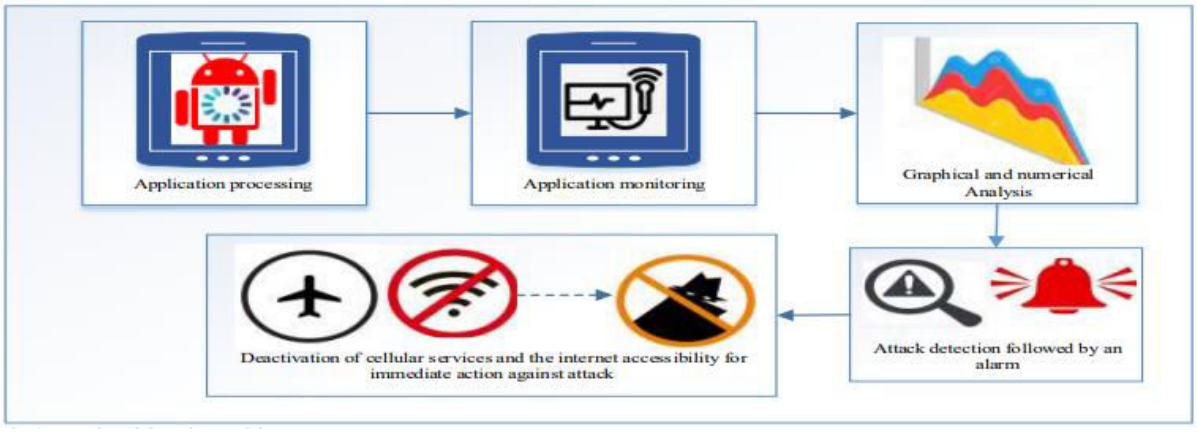

Figure 3: Apps based detection model

The detection model carries out the schematic of intrusion detection for application-based attack. During the invalid application processing in the memory and database of the cellular smartphone, the monitoring of each application is observed using ABMA. The mechanism of ABMA is performed based on the power consumption, battery level, and data usage. Other parameters such as energy efficiency, complexity, battery consumption are parameters that

can be considered for the analysis. The ABMA in absence of the intruder acts as a threshold for intruder app detection. The threshold is adaptively obtained from the live data of applications. From the comparative analysis of the ABMA pattern, the normal and the invalid installed application is detected. The attack detection is followed by the alarm or the pop up notification of the invalid app activity. For immediate protection against the attack, the cellular services and the internet accessibility (WiFi, mobile data, BlueTooth) are turned off. The procedure of deactivation of services is informed to the user with the popup notification. The process flow of the intruder app detection mechanism is shown in Fig. 3.

Apps based intrusion detection using ABMA

The radiation pattern demonstrates the comparative Application-based Behavioral Model Analysis (ABMA) for the apps based intrusion detection. The behavioral model determines the apps based peak power usage corresponding to the data and battery level. Demonstrates the 3D ABMA pattern for the 1-hour duration using WiFi access in absence of the intruder. A comparable rise in the power, data, and the decrease in battery level is observed in presence of the intruder. The power, data, and the battery level incorporated by the intruder app for its processing ultimately affect the overall usage. Therefore, intruder app processing consumption forms the basis of the proposed methodology of intrusion detection.

Further, for the one day and one week duration, the strategy of comparative ABMA is followed for intrusion detection. The comparative corresponding intruder app processing consumption forms the criterion of the intrusion detection. A minor processing activity by the intruder app leads to the change in the final value of the ABMA. Table IV depicts the ABMA parameter analysis using the combination of both the mobile data and the WiFi for the internet access of the app processing.

IV. Results and Evaluation

The apps based energy efficiency analysis with respect to the number of active applications. It has been observed that with an increase in the number of active applications, the apps based energy efficiency tends to increase thereby, showing a direct relationship with active applications of the smartphone. Thus, more applications are executed with the same amount of energy. In presence of the intruder, the apps based energy efficiency reduces contrary to the valid application-based energy efficiency. The presence of the intruder increases the comparable energy consumption thereby deteriorating the energy efficiency. However, the

impact of the intruder is dominant for the one day duration as compared to the one hour duration. The effect of the intruder app on the parameter of energy consumption. From the analysis, it is observed that with an increase in the number of active applications, energy consumption tends to increase. The presence of the intruder has a drastic impact on overall energy consumption. The intruder processing increases the requirement of energy. Consequently, creates a rise in overall energy consumption. Thus, the increase in the app-based energy consumption in the presence of the intruder can be used as the criterion for app-based intrusion detection. It is clearly detected that the influence of intruder applications is majorly at the longer durations. The presence of the intruder increases the utilization of the available resources in the form of energy, CPU usage, current drawn, data, and power. In the presence of more number of active applications more is the processing requirement. Therefore, more is the complexity. Moreover, resource utilization has a direct impact on the duration. Resource utilization increases with an increase in the duration of the active applications. Thus, increases the overall app-based complexity.

V. Limitations

The proposed model can be extended using artificial intelligence associated with the mechanisms of reward and punishment such as reinforcement learning algorithms. Artificial intelligence can prove effectual in terms of accuracy, precision, and reliability. However, the introduction of artificial intelligence in the proposed model increases the requirement of high-end processors and more complexity. The proposed behavioral model can prove effective in specific applications such as net banking applications, online shopping applications, social networking applications, stock market applications and other applications that require preeminent confidentiality.

VI. Conclusion

The increase in the use of applications on the smartphone has enhanced numerous vulnerabilities and threats in the form of loss in confidentiality, invalid access control permissions, and invalid authorizations, links to vulnerable sources. In this paper, an application-based attack modeling and attack detection is proposed to address such challenges. The attack modeling incorporates the end-user vulnerable application installation on the smartphone. The possible installation integrates hidden visibility activation mode to process the mechanism. The detection process evaluates ABMA scheme for the invalid application entry. The application-based analysis is estimated using power consumption,

battery level, and data usage. The comparative analysis is observed for application intrusion detection. For the immediate countermeasure of the attack, an alarm is raised followed by the disconnection of cellular services and internet accessibility.

VII. References

[1] M. S. Abdalzaher and O. Muta, "A Game-Theoretic Approach for Enhancing Security and Data Trustworthiness in IoT Applications," in IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11250-11261, Nov. 2020.

[2] C. Shen, Y. Chen, Y. Liu and X. Guan, "Adaptive Human–Machine Interactive Behavior Analysis With Wrist-Worn Devices for Password Inference," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 12, pp. 6292-6302, Dec. 2018.

[3] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han and X. Zhang, "Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection," in IEEE Transactions on Information Forensics and Security, vol. 9, no. 11, pp. 1869-1882, Nov. 2014.

[4] Z. Lu, W. Wang and C. Wang, "Modeling, Evaluation and Detection of Jamming Attacks in Time-Critical Wireless Applications," in IEEE Transactions on Mobile Computing, vol. 13, no. 8, pp. 1746-1759, Aug. 2014.

[5] J. Mao, S. Zhu, X. Dai, Q. Lin and J. Liu, "Watchdog: Detecting Ultrasonic-Based Inaudible Voice Attacks to Smart Home Systems," in IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8025-8035, Sept. 2020.

[6] L. Wu, X. Du and X. Fu, "Security threats to mobile multimedia applications: Camera-based attacks on mobile phones," in IEEE Communications Magazine, vol. 52, no. 3, pp. 80-87, March 2014.

[7] R. Spreitzer, V. Moonsamy, T. Korak and S. Mangard, "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," in IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 465-488, Firstquarter 2018.

[8] A. Maiti, M. Jadliwala, J. He and I. Bilogrevic, "Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches," in IEEE Transactions on Mobile Computing, vol. 17, no. 9, pp. 2180-2194, 1 Sept. 2018.

[9] S. Naval, A. Pandey, S. Gupta, G. Singal, V. Vinoba and N. Kumar, "PIN Inference Attack: A Threat to Mobile Security and Smartphone-controlled Robots,"in IEEE Sensors Journal, 2021. doi: 10.1109/JSEN.2021.3080587.

[10] J. Yu, L. Lu, Y. Chen, Y. Zhu and L. Kong, "An Indirect Eavesdropping Attack of Keystrokes on Touch Screen through Acoustic Sensing," in IEEE Transactions on Mobile Computing, vol. 20, no. 2, pp. 337-351, 1 Feb. 2021.

[11] Y. Zhang, M. Yang, G. Gu and H. Chen, "Rethinking Permission Enforcement Mechanism on Mobile Systems," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 10, pp. 2227-2240, Oct. 2016.

[12] F. Roesner, "Designing Application Permission Models that Meet User Expectations," in IEEE Security & Privacy, vol. 15, no. 1, pp. 75-79, Jan.- Feb. 2017.

[13] B. Rashidi, C. Fung, A. Nguyen, T. Vu and E. Bertino, "Android User Privacy Preserving Through Crowdsourcing," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, pp. 773-787, March 2018.

[14] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an and H. Ye, "Significant Permission Identification for Machine-Learning-Based Android Malware Detection," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3216-3225, July 2018.

[15] J. Huang, Y. Xiong, W. Huang, C. Xu and F. Miao, "SieveDroid: Intercepting Undesirable Private-Data Transmissions in Android Applications," in IEEE Systems Journal, vol. 14, no. 1, pp. 375-386, March 2020.

[16] K. Gai, M. Qiu and H. Zhao, "Privacy-Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing," in IEEE Transactions on Big Data, vol. 7, no. 4, pp. 678-688, 1 Sept. 2021.

[17] H. Fu et al., "Towards Automatic Detection of Nonfunctional Sensitive Transmissions in Mobile Applications," in IEEE Transactions on Mobile Computing, 2020, doi: 10.1109/TMC.2020.2992253.

[18] Y. Zhang et al., "Looking Back! Using Early Versions of Android Apps as Attack Vectors," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 2, pp. 652-666, 1 March-April 2021.

[19] C. Ma, T. Wang, L. Shen, D. Liang, S. Chen and D. You, "Communication-based attacks detection in android applications," in Tsinghua Science and Technology, vol. 24, no. 5, pp. 596-614, October 2019.

[20] K. Chen, Y. Zhang and P. Liu, "Leveraging Information Asymmetry to Transform Android Apps into Self-Defending Code Against Repackaging Attacks," in IEEE Transactions on Mobile Computing, vol. 17, no. 8, pp. 1879-1893, 1 Aug. 2018.

[21] L. Wu, X. Du and J. Wu, "Effective Defense Schemes for Phishing Attacks on Mobile Computing Platforms," in IEEE Transactions on Vehicular Technology, vol. 65, no. 8, pp. 6678-6691, Aug. 2016.

[22] H. Mustafa, W. Xu, A. Sadeghi and S. Schulz, "End-to-End Detection of Caller ID Spoofing Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 3, pp. 423-436, 1 May-June 2018.

[23] H. Ma, S. Li, D. Gao, D. Wu, Q. Jia and C. Jia, "Active Warden Attack: On the (In)Effectiveness of Android App Repackage-Proofing," in IEEE Transactions on Dependable and Secure Computing, July 2021, doi: 10.1109/TDSC.2021.3100877.

[24] H. Ma, S. Li, D. Gao and C. Jia, "Secure Repackage-Proofing Framework for Android Apps using Collatz Conjecture," in IEEE Transactions on Dependable and Secure Computing, June 2021, doi: 10.1109/TDSC.2021.3091654.