Behavioral Model for Live Detection of Apps Based Attack

Misbah Shafi, Student Member, IEEE, Rakesh Kumar Jha®, Senior Member, IEEE, and Sanjeev Jain®, Senior Member, IEEE

Abstract—Smartphones with the platforms of applications are gaining extensive attention and popularity. The enormous use of different applications has paved the way for numerous security threats. The threats are in the form of attacks such as permission control attacks, phishing attacks, spyware attacks, botnets, malware attacks, and privacy leakage attacks. Moreover, other vulnerabilities include invalid authorization of apps, compromise on the confidentiality of data, and invalid access control. In this article, an application-based attack modeling and attack detection are proposed as a novel attack vulnerability is identified based on the app execution on the smartphone. The attack modeling involves an end-user vulnerable application to initiate an attack. The vulnerable application is installed at the background end on the smartphone with hidden visibility from the end-user, thereby accessing the confidential information. The detection model involves the proposed technique of an application-based behavioral model analysis (ABMA) scheme to address the attack model. The model incorporates application-based comparative parameter analysis to perform the process of intrusion detection. The ABMA is estimated by using the parameters of power, battery level, and data usage. Based on the source Internet accessibility, the analysis is performed using three different configurations, Wi-Fi, mobile data, and a combination of the two. The simulation results verify and demonstrate the effectiveness of the proposed model.

Index Terms—Application-based behavioral model analysis (ABMA), attack modeling, energy consumed, intrusion detection, security, smartphone applications.

I. INTRODUCTION

A. Motivation

In recent years, smartphone 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 threats, wherein interested applications 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 smartphone. 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 smartphones [6]. The side-channel attack exploits the leakage data to limit the data confidentiality on smartphones [7], [8]. Pin inference attack is identified as a privacy threat for the devices controlled by smartphones [9]. Indirect eavesdropping attack is another possible menace that makes use of acoustic sensing to execute the attack on the smartphone [10].

Permission control is one of the primary countermeasures against the possible security risks in smartphones. 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 crowdsourcing [13], and significant permission identification (Sig PID) [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.

The data privacy control scheme is another security enhancement technique against application attacks. Seivedroid is the privacy control technique to mark objectionable and confidential data [15]. Privacy-preserving data encryption of applications involves selective data encryption with the association of time constraints [16]. Flow intent is defined as the identification of nonfunctional confidential data transmission and prevents suspicious data transmissions from application visual interfaces [17]. However, locating sensitive data is one of the major challenges evolved in these mechanisms. Speed and adaptability are the other issues that limit the security enhancement of smartphone applications.

The overall constraints of the conventional security enhancement schemes of smartphone applications include particular application specifications-based security enhancement
TABLE I
COMPARISON OF RECENT SCHEMES WITH THE PROPOSED SCHEME

| Ref | Objective | Attack type | Applied application | Technique/tool | Optimized parameters | Motivation |
|-----|-----------|-------------|---------------------|----------------|----------------------|------------|
| [18] | Earlier application version as attack threat | Man in the middle attack | Facebook, Sina weibo, qihoo 360 | Driodskynet | Success rate, protection, similarity index of early and upgraded versions | Disadvantages of [18]-[24]: Application specific, attack, technology dependent, not applicable to latest or upcoming applications, inadaptive |
| [19] | Communication based attack detection using intent abnormality analysis | Communicatio based attack, hijacking attack | 57 real world applications | Component level data flow analysis | Detection rate, time overhead, space overhead | |
| [20] | Self defending mechanism to let the repackaged application expose themselves automatically | Reuse attacks / repackaging attacks | RandomMusic, standap timer, angle, asquare, swiFTP, pdfview, opensnudoku, achatengine, wordpress, Apidemos | Self defending coding technique | Performance overhead, diversity test | |
| [21] | Antiphishing scheme for mobile platforms | Phishing attack | Facebook, paypal, ebay | MobiFish | Execution time | |
| [22] | End to end caller ID verification scheme | Caller ID spoofing attack | Caller Dec application | CallerDec | Accuracy, time, voltage usage, precision, recall | |
| [23] | Active warden attack is detected using integrity and multiparty verification framework | Active warden attack | Instagram, frozen bubble | Multiparty verification | Average overhead, code bloat | |
| [24] | Repackage proofing framework for detection and response stage | Privacy leakage attack | Frozen bubble | Android data flow analyzer | Average overhead, code bloat | |
| Proposed | Mobile application-based attack model and detection model for mobile application | Mobile application-based attacks | Facebook, whatsapp, gmail, amazon, youtube, chrome | Application thermal pattern analysis | Energy efficiency, power usage, battery usage, data | |

B. Related Work

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 Play Store. In [19], the possible security menaces are located in the Android operating system having intercomponent 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], an 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 the validity and effectiveness of the mechanism.

C. Novelty

The application-based behavioral model analysis (ABMA) is a novel methodology defined for the security improvement of smartphone platforms. Conventional schemes of security enhancement mechanisms in smartphones are attack-specific or application-specific. A generalized security scheme independent of versions and type of application is yet to be addressed. Also, the reliability with upgradation and optimized statistical parameters requires immense attention. The behavioral model for smartphone-based applications is an innovative initiative to address these challenges with an effectual performance. The model is independent of the technology and the updates of the applications of the smartphone. It provides the live detection app-based attack with adaptive capability. The detection model ensures the app-based attack detection on authorization, confidentiality, and integrity with an efficient and less complex methodology. The significant contribution of this article is given as follows.

1) A novel and probable application-based attacking model has been identified with an efficient strategy of hidden access installation in the background and the hidden visibility from the end-user.

2) The detection model for the applications of the smartphone has been proposed to address the modeled attack.
The evaluation is based on the comparative analysis of the behavioral model such that the actual parameters are compared with the parameters in the presence of the intruder application.

3) To counteract the detected intrusion, an alarm is raised as the immediate response followed by the disconnection of the cellular services and Internet accessibility.

4) The obtained results illustrate that the proposed scheme can prove an effective mechanism using ABMA in terms of power, data, and battery level.

D. Organization

The article is organized as follows. Section I presents the introduction. Section II describes the system model and problem formulation. The realization and representation of the proposed methodology are discussed in Section III. An illustrative example of the proposed methodology is demonstrated in Section IV. The performance evaluation is analyzed in Section V. To the closure of the article, the conclusion is given in Section VI.

II. System Model and Problem Formulation

In this section, the system model and the characteristics model are illustrated for a detailed network description. Furthermore, the problem formulation is discussed for the proposed scheme.

A. System Model

The fundamental of the system model consists of the end-user service application-based scenario. The overall background communication network involves a BS and \( n \) number of users. The end-user connectivity services are assumed to be provided by any of the 5G/6G technologies. The \( i \)th user is considered as the target of attack and, therefore, analyzed for attack modeling and attack detection. The end-user is portrayed in the form of a smartphone having \( n \) number of different applications, where \( n \in \mathbb{N} \). Consider \( A = A_1, A_2, A_3, \ldots, A_n \) as the number of availed applications on the smartphone. The intruder is considered to be present in the scenario having an interest in obtaining the confidential information of the valid user from the smartphone. The intruder applies advertisement access add on the other applications to breach through the smartphone validations. The intruder is assumed to have a keen watch on the actions of the valid user on the smartphone. Under any immediate action of the advertisement add, the link of advertisement follows the downloading application, such as Play Store for the download and installation process such that

\[
A = \{A_1, A_2, A_3, \ldots, A_n, A_{n+1}\}. \tag{1}
\]

The \( A_{n+1} \) application processes in the hidden visibility mode to prevent the detection. The applications \( A = A_1, A_2, A_3, \ldots, A_n \) are present in the list of the smartphone. The downloaded applications are more prone to vulnerable breaches and have more compromised security compared to the system applications. Therefore, we consider the downloaded application as the source link of the vulnerability, advertised on it.

**Proposition 1:** The source background communication connectivity scenario is considered free from intruders, is completely validated, and, thus, provides secure Internet access with good network coverage from the back end.

B. Characteristic Parameter Model

The parameter of power is one of the parameters that determine the performance of the device. The execution of the attack affects the overall performance and, thereby, increases the power usage. The overall consumed power \( P_e \) of the smartphone involves the combination of the static \( P_s \) and dynamic power \( P_d \) for the time instant \( t = 0^+ \) to \( t = T \), which is given by

\[
P_e(t) = P_d(t) + P_s(t) \tag{2}
\]

adding the impact of corresponding leakage power. Equation (2) can be rewritten as

\[
P_e(t) = P_d(t) + P_s(t) + P_w(t) + P_{sl}(t)
\]

where \( P_w \) is the leakage power in the dynamic state and \( P_{sl} \) is the leakage power in the static state.

**Definition 1:** The total power consumed in applications \( P_{DA} \) is the sum of the power consumed by the individual applications for the time instant \( t = 0^+ \) to \( t = T \). The attack via an application affects the total power consumed by the applications.

Using (1), the total power consumed for the corresponding applications can be defined as

\[
P_{DA}(t) = P_{A_1}(t) + P_{A_2}(t) + P_{A_3}(t) + \cdots + P_{A_n}(t) + P_{sl}(t). \tag{4}
\]

**Proposition 2:** For the case of simplicity, the circuit power \( P_{c_s} \) is excluded from the analysis such that solely power consumed for the communication is considered.

Based on the demand of the applications and the resource allocation of the corresponding application, the total estimated power for the applications can be obtained as

\[
P'_{DA}(t) = \sum_{i=1}^{n} P_{A_i}(t) \tag{5}
\]

or

\[
P'_{DA}(t) = \sum_{i=1}^{n} \frac{\sigma(t)}{|h_{A_i}(t)|^2} \left[ 2^{\frac{\gamma(t)}{\beta}} - 1 \right] \tag{6}
\]

where \( \sigma \) is the noise power, \( h \) is the channel coefficient, \( \beta \) is the allocated bandwidth, and \( \gamma \) is the data rate of the corresponding application. \( P_{A_i} \) is the power consumed by the \( i \)th application.

**Definition 2:** The total data rate is the sum of data rates of individual applications. The presence of an attack has a direct impact on the usage of the data.

Using (1), the corresponding total data rate \( \gamma_A(t) \) of the applications for the time instant \( t \) is given by

\[
\gamma_A(t) = \sum_{i=1}^{n} \gamma_{A_i}(t) \tag{7}
\]
\[
\gamma_A(t) = \sum_{i=1}^{n} \beta_i(t) \log_2(1 + \varphi_i(t)) \tag{8}
\]

where \( \beta_i \in \{\beta_{A1}, \beta_{A2}, \beta_{A3}, \ldots, \beta_{An}\} \) is the corresponding bandwidth allocated to the application and \( \varphi \) is the signal-to-noise ratio (SNR).

**Definition 3:** The total battery consumed in applications is the sum of battery consumed by the individual applications

\[
B_A(t = T) = \sum_{i=1}^{n} [B_{A_i}(t = 0^+) - B_{A_i}(t = T)] \tag{9}
\]

where \( B_A(t = T) \) defines the total consumed battery level by the applications, \( B_{A_i}(t = 0^+) \) is the initial battery level for the \( i \)th application, and \( B_{A_i}(t = T) \) is the final battery level at the time instant \( T \) for the \( i \)th application. The battery lifetime [23] can be evaluated as

\[
L_b(t) = \frac{B_i(t)}{\sum_{i=1}^{n} I_{A_i}(t)} \cdot 0.7 \text{(in hours)} \tag{10}
\]

where \( L_b \) is the total battery lifetime, \( B_i \) is the total battery capacity available, and \( I_{A_i} \) is the total current drawn by the \( i \)th application. The value of 0.7 accounts for external factors.

**Definition 4:** The total energy efficiency provided by the smartphone for the applications is the sum of the individual energy efficiencies of the corresponding applications. Similarly, the presence of the attack affects energy efficiency.

For each of the applications, the total energy efficiency of the smartphone for the applications is given by

\[
EE_A(t) = \sum_{i=1}^{n} EE_{A_i}(t). \tag{11}
\]

Equation (11) can be obtained as

\[
EE_A(t) = \sum_{i=1}^{n} \left( \frac{\gamma_{A_i}(t)}{P_{A_i}(t)} \right). \tag{12}
\]

These parameters critically define the characteristic behavior of the applications and eventually affect the performance of the smartphone.

**C. Problem Formulation**

The proposed scheme identifies the possible attack modeling and intrusion detection based on the characteristic parameters of data rate, energy efficiency, power, and battery. The prime aim is to detect the application-based intrusion using the analysis of these parameters. The collective performance of these parameters defines the criteria as

\[
P_{cs}(t) \approx P_{obs}(t) \tag{13}
\]

such that

\[
P_{obs}(t) \approx (P_{A1}(t) + P_{A2}(t) + P_{A3}(t) + \cdots + P_{An}(t)). \tag{14}
\]

The corresponding criteria of data rate for intrusion detection are defined as

\[
\gamma_{cs}(t) \approx \gamma_{obs}(t). \tag{15}
\]

**Pseudo Code 1:** Behavioral Model Analysis

1: **Input parameters**
   - Battery usage
   - Battery status
   - Data rate
   - Power consumption status
   - Energy consumption status
   - Number of applications
   - Operating bandwidth \( \beta \), Noise power \( \sigma \)

2: **Initialization**
   - Initialize time \( t \)
   - Set of applications \( A = \{A_1, A_2, A_3, \ldots, A_n\}, n \in \mathbb{N} \)
   - Circuit power consumption \( P_{cA} \)
   - Initial battery level in terms of battery lifetime \( L_b \)
   - Initialize power consumption of applications
   - Initialize energy consumption of applications
   - Initialize data rate of applications
   - Initialize battery usage of applications

3: **Intruder application detection using behavioral model analysis**
   - Estimated analysis for time instant \( t \)
     - for \( i = 1 : n \) \( i \neq n \) is the number of applications
     - Determine power consumed by the \( i \)th application \( P_{A_i}(t) = \{P_{A_i}\} \)
     - Determine data rate of the \( i \)th application \( \gamma_{A_i}(t) = \{\gamma_A\} \)
     - Determine battery usage of the \( i \)th application \( B_{A_i}(t) = \{B_A\} \)
     - end
     - Compute total power consumed by \( n \) applications
       \[
P_{dA}(t) = \sum_{i=1}^{n} P_{A_i}(t)
\]
     - Compute total data rate consumed by \( n \) applications
       \[
\gamma_A(t) = \sum_{i=1}^{n} \gamma_{A_i}(t)
\]
     - Compute total battery consumed by \( n \) applications
       \[
B_A(t = T) = \sum_{i=1}^{n} [B_{A_i}(t = 0^+) - B_{A_i}(t = T)]
\]

4: **Observed analysis for time instant \( t \)**
   - find
     - Total power consumed, data rate consumed, battery consumed by the applications \( P_{dA}(t), \gamma_A(t), B_A(t) \)
     - if \( P_{dA}(t) \neq P_{obs}(t), \gamma_A(t) \neq \gamma_{obs}(t), B_A(t) \neq B_{obs}(t) \)
       - Intruder application is present
       - Raise an alarm
       - Disconnect the cellular and internet accessibility
     - else
       - Secure mobile computing is processing
     - end if
   - **wait for time instant \( \delta t \)** \( \delta t \) is the very small time instant such that constant monitoring is maintained

4: **Update in the number of applications**
   - if \( n(t) \neq n(t + \delta t) \)
     - update \( n \)
   - end if

Similarly,

\[
P_{es}(t) \approx P_{obs}(t). \tag{16}
\]
Also, \[ E_{es}(t) \approx E_{obs}(t) \] (17)
where \( P_{obs}, \gamma_{obs}, B_{obs}, \) and \( E_{obs} \) are the sum of the individual application parameters observed. \( P_{es}, \gamma_{es}, B_{es}, \) and \( E_{es} \) are the total estimated application parameters. Constraints are given in (13) and (15)–(17) to confirm the validation of the applications while allocating the services. It defines the collective performance of the applications such that the estimated boundary-bound value of the parameters is not exceeded.

III. REALIZATION AND REPRESENTATION

This section presents the realization and representation of the proposed mechanism by depicting Pseudocode 1. Pseudocode 1 provides a complete insight into the execution and procedure of intruder application detection. The \( n \) valid applications such that \( n \in \mathbb{N} \) for time instant \( t \) are considered for the behavioral model analysis depicting power, battery, and data rate parameters. The parameters are analyzed based on the performance of individual applications. The estimated parameters in the absence of the attack act as a threshold for the detection of the intruder application. Similarly, the behavioral model analysis using these parameters in the presence of the intruder application is evaluated, forming the observed analysis. The estimated behavioral model in the absence of the attack is compared with the observed behavioral model in the presence of the attack to detect the intruder application. Furthermore, for secure mobile computing in the absence of an attack, the estimated behavioral model is equivalent to the observed behavioral model. The attack model and the detection model are well shown in Figs. 2 and 3.

IV. PROPOSED METHODOLOGY OF APPLICATION-BASED ATTACK DETECTION: HOW IT WORKS

This section defines the proposed methodology of the application-based attack model and the detection model.

A. Attack 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 app-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. 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. 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.

Illustration: In the absence of the attack, the set of applications is defined as \[ A = \{A_1, A_2, A_3, \ldots, A_n\}. \] (18)
The corresponding parameters for different applications are given in (5), (7), (10), and (11). In the presence of the attack, the set of applications is defined in (1).

B. Application Intrusion Detection Model

The detection model carries out the schematic of intrusion detection for application-based attacks. 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, and battery consumption, are parameters that can be considered for the analysis. The ABMA in the 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 invalid installed applications are 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 (Wi-Fi, mobile data, and 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.

**Illustration:** The parameters in the absence of attack are considered as the threshold for intrusion detection. The behavioral model characteristics for the intrusion detection can be defined as

\[ P_A^{\text{att}}(t) > P_A^{\text{val}}(t). \]  

(19)

Also,

\[ B_A^{\text{att}}(t) < B_A^{\text{val}}(t) \]  

(21)

and

\[ E_A^{\text{att}}(t) > E_A^{\text{val}}(t) \]  

(22)

where \( P_A^{\text{att}} \) is the total power consumed by the applications in the presence of the attack, \( P_A^{\text{val}} \) is the total power consumed by the applications in the absence of the attack. \( \gamma_A^{\text{att}} \) is the total data rate of the applications in the presence of the attack. \( \gamma_A^{\text{val}} \) is the total data rate of the applications in the absence of the attack. \( B_A^{\text{att}} \) is the battery level in the presence of the attack, and \( B_A^{\text{val}} \) is the battery level in the absence of the attack. \( E_A^{\text{att}} \) is the total energy consumption by the applications in the presence of the attack, and \( E_A^{\text{val}} \) is the total energy consumption by the applications in the absence of the attack (refer to Lemma 1).
V. SIMULATION ANALYSIS

This section provides complete insight into the analytical results attained by the real-time investigation of the app-based attack. The scenario involves the app-based analysis on the smartphone. The specifications of the analysis parameters are shown in Table II. The section is divided into two parts. The first defines the ABMA, and the second part evaluates the results.

A. App-Based Intrusion Detection Using ABMA

The radiation pattern demonstrates the comparative ABMA for app-based intrusion detection. The behavioral model determines the app-based peak power usage corresponding to the data and battery level. Fig. 4(a) demonstrates the 3-D ABMA pattern for the 1-h duration using Wi-Fi access in the absence of the intruder. A comparable rise in the power, data, and the decrease in battery level is observed in the presence of the intruder. The power, data, and the battery level incorporated by the intruder app for its processing ultimately affect the overall usage, as shown in Fig. 4(b). Therefore, intruder app processing consumption forms the basis of the proposed methodology of intrusion detection. Furthermore, for the one-day and one-week duration, the strategy of comparative ABMA is followed for intrusion detection, as shown in Fig. 4(c)–(f).

TABLE II

| Parameter                              | Value                           |
|----------------------------------------|---------------------------------|
| Number of valid applications for analysis | 6                               |
| Maximum battery capacity               | 4000mAh                         |
| WiFi link speed                        | 72Mbps                          |
| Number of intruder application         | 1                               |
| Mobile Type                            | android                         |
| Smartphone specifications              | Redmi Xiaomi, Note 7 Pro       |
| Mobile brightness                      | 50%                             |
| Reading mode                           | off                             |
| Android version                        | 10 QKQ1.190915.002              |
| Battery status                         | Not charging                     |
| Average mobile data speed              | 1 Mbps                          |
Table III provides a complete understanding of the ABMA using six different applications using mobile data as the source of Internet access for the apps on the smartphone. The comparative corresponding intruder app processing consumption forms the criterion of 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 Wi-Fi for the Internet access of the app processing.

### B. Results and Implications

Fig. 5 illustrates the app-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 app-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 the presence of the intruder, the app-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 1-h duration.

Fig. 6 represents 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 and, 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 and, thus, increases the overall app-based complexity. The complexity in terms of the percentage increase is depicted in Fig. 7. The complexity in terms of power usage is shown in Fig. 8.

Fig. 9 provides a clear understanding of the battery lifetime with respect to the increase in the number of active applications of the smartphone. It is illustrated in the figure that the battery lifetime worsens in the presence of the intruder app. The activation of the intruder app processing upsurges the battery consumption ultimately affecting the overall battery lifetime.

Fig. 10 shows the average app-based energy efficiency under six active apps. Different sources of Internet access are compared to demonstrate the parameter of overall energy efficiency with and without the influence of the intruder. The Wi-Fi-based Internet source for the apps’ processing is perceived to have the maximum apps-based overall energy efficiency. Based on the duration, the overall one-week duration provides the maximum app-based energy efficiency. Hybrid access designates...
Fig. 8. Complexity with respect to the number of active applications.

Fig. 9. Battery lifetime with respect to the number of active applications.

The alternate combination of Wi-Fi Internet access and mobile data-based Internet access. Hybrid access provides higher app-based energy efficiency compared to mobile data access. A radical difference is observed between the presence and absence of the intruder app.

C. 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 effective in terms of accuracy, precision, and reliability. However, the introduction of artificial intelligence in the proposed model increases the requirement for 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 of confidentiality, invalid access control permissions, and invalid authorizations, links to vulnerable sources. In this article, application-based attack modeling and attack detection are proposed to address such challenges. The attack modeling incorporates the end-user vulnerable application installation on the smartphone. The possible installation integrates a hidden visibility activation mode to process the mechanism. The detection process evaluates the A B M A 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 an immediate countermeasure to the attack, an alarm is raised followed by the disconnection of cellular services and Internet accessibility.

APPENDIX

Lemma 1: The constraints defined in (19)–(22) in terms of power, battery, data rate, and energy specify optimal detection of the vulnerable attack application. The criterion for the application-based attack detection is given as follows:

1) $P_{\text{att}}^A(t) > P_{\text{val}}^A(t)$
2) $\gamma_{\text{att}}^A(t) > \gamma_{\text{val}}^A(t)$
3) $B_{\text{att}}^A(t) < B_{\text{val}}^A(t)$
4) $E_{\text{att}}^A(t) > E_{\text{val}}^A(t)$.

Proof:

i. We note from (5) that, in the absence of the intruder application, the total power consumed by the valid applications
can be estimated as
\[ P_A^{\text{val}}(t) = \sum_{i=1}^{n} P_{Ai}(t). \] (23)

Now, in the presence of the intruder application, draw some amount of power to execute the processing such that
\[ P_A^{\text{att}}(t) = \sum_{i=1}^{n} P_{Ai}(t) + P_{A(n+1)}^{\text{att}}(t). \] (24)

Using (23) in (24), we get
\[ P_{A(n+1)}^{\text{att}}(t) = P_A^{\text{val}}(t) - P_A^{\text{att}}(t). \] (26)

From (25) and (26), it is concluded that
\[ P_{A(n+1)}^{\text{att}}(t) > P_A^{\text{att}}(t). \] (27)

ii. Part ii follows a similar to that of part i. Using (7), in the absence of the attack, the total data rate of the valid application for time instant \( t \) is obtained as
\[ \gamma_A^{\text{val}}(t) = \sum_{i=1}^{n} \gamma_{Ai}(t). \] (28)

Equation (28) is also represented as
\[ \gamma_A^{\text{val}}(t) = \sum_{i=1}^{n} \beta_i(t) \log_2(1 + \varphi_i(t)). \] (29)

As per the intruder application requirements, the resource allocation takes place such that the total data rate of the applications in the presence of the intruder application is estimated as
\[ \gamma_A^{\text{att}}(t) = \gamma_A^{\text{val}}(t) + \beta_{n+1} \log_2(1 + \varphi_{n+1}(t)). \] (30)

Equation (30) can be deduced as
\[ \gamma_A^{\text{att}}(t) > \gamma_A^{\text{val}}(t). \] (31)

iii. Part iii specifies the battery evaluation to detect the attack. Using (9), the total consumed battery for \( n \) valid applications is given as
\[ B_A(t = T) = \sum_{i=1}^{n} [B_{Ai}(t = 0^+) - B_{Ai}(t = T)]. \] (32)

Let
\[ \sum_{i=1}^{n} [B_{Ai}(t = 0^+) - B_{Ai}(t = T)] = |Z(t)|_{t=0}^{t=T}. \] (33)

For the intruder application, the consumed battery for time instant \( t \) is denoted as
\[ B_{A(n+1)}^{\text{att}}(t = T) = [B_{A(n+1)}(t = 0^+) - B_{A(n+1)}(t = T)]. \] (34)

Let
\[ |B_{A(n+1)}(t = 0^+) - B_{A(n+1)}(t = T)| = |Y_{n+1}(t)|_{t=0}^{t=T}. \] (35)

Noting from (10), the battery life time in the absence of the attack is given as
\[ L_b(t) = \frac{B_{c}(t)}{\sum_{i=1}^{n} I_{Ai}(t)} \approx 0.7 \text{ (in hours)}. \] (36)

The remaining battery lifetime (battery level) after the use of \( n \) applications for time \( T \) can be interpreted as
\[ L_b'(t) = \frac{B_{c}(t) - |Z(t)|_{t=0}^{t=T} - B_{c}\text{tot}}{\sum_{i=1}^{n} I_{Ai}(t) + \sum_{i=1}^{n} I_{Ai(n+1)}(t)} \approx 0.7 \text{ (in hours)}. \] (37)

where \( B_{c}\text{tot} = C \) is the battery consumed in processing other than applications and is considered a constant; \( I_{c}\text{tot} = D \) is the current drawn in other processing. Equation (37) can be denoted as
\[ L_b'(t) = \frac{B_{c}(t) - |Z(t)|_{t=0}^{t=T} - C - |Y_{n+1}(t)|_{t=0}^{t=T}}{\sum_{i=1}^{n} I_{Ai}(t) + \sum_{i=1}^{n} I_{Ai(n+1)}(t)} \approx 0.7 \text{ (in hours)}. \] (39)

Comparing (37) and (39), we get
\[ L_b'(t) > L_b''(t). \] (40)

Correspondingly, from (40), it can be deduced as
\[ B_A^{\text{att}}(t) > B_A^{\text{val}}(t). \] (41)

iv. The energy parameter analysis specifies intruder application detection. Noting from (12) in the absence of the intruder application, the overall energy efficiency is given as
\[ EE_{\text{Tot}}^{\text{val}}(t) = \frac{\gamma_{\text{Tot}}^{\text{val}}(t)}{P_{\text{Tot}}^{\text{val}}}. \] (42)

Now, in the presence of the intruder application, the energy efficiency is obtained as
\[ EE_{\text{Tot}}^{\text{att}}(t) = \frac{\gamma_{\text{Tot}}^{\text{val}}(t) - \gamma_{\text{Tot}}^{\text{att}}(t)}{P_{\text{Tot}}^{\text{val}} + P_{\text{Tot}}^{\text{att}}}. \] (43)

Comparing (42) and (43), we get
\[ EE_{\text{Tot}}^{\text{val}}(t) > EE_{\text{Tot}}^{\text{att}}(t). \] (44)

where \( P_{\text{Tot}}^{\text{val}} \) is the total power consumed, \( \gamma_{\text{Tot}}^{\text{val}}(t) \) is the total available data rate, \( \gamma_{\text{Tot}}^{\text{att}}(t) \) is the allotted data rate to the intruder application, and \( P_{\text{Tot}}^{\text{att}} \) is the power consumed in intruder application.

Similarly, the energy consumed for \( n \) applications in the absence of the attack can be given as
\[ E_T^{\text{val}}(t) = \frac{E^{\text{Bat}}(t)L}{3600} \text{ Watt hr}. \] (45)

where \( E^{\text{Bat}}(t) \) is the battery consumed and \( L \) is the transmission latency. In the presence of the intruder application, the energy consumed is defined as
\[ E_T^{\text{att}}(t) = \frac{E^{\text{Bat}}_A(t) + E^{\text{Bat}}_{A(n+1)}(t)}{3600} L \text{ Watt hr}. \] (46)
Using (45) in (46), we get

\[ E_{att}^T(t) = E_{val}^T(t) + \frac{(E_{Bat}^{A(n+1)}(t) L}{3600} \text{ Watt hr.} \]  \hspace{1cm} (47)

From (47), we get

\[ E_{att}^T(t) > E_{val}^T(t) \]  \hspace{1cm} (48)

where \( E_{Bat}^{A(n+1)}(t) \) is the battery consumed by the intruder application.

■ Few examples of real-time app analysis are shown in Figs. 11 and 12.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support provided by 5G and IoT Lab, DoECE, and TBIC, Shri Mata Vaishno Devi University, Katra, Jammu.

REFERENCES

[1] M. S. Abdalzaher and O. Muta, “A game-theoretic approach for enhancing security and data trustworthiness in IoT applications,” IEEE Internet Things J., 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,” IEEE Trans. Neural Netw. Learn. Syst., 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,” IEEE Trans. Inf. Forensics 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,” IEEE Trans. Mobile Comput., 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,” IEEE Internet Things J., vol. 7, no. 9, pp. 8025–8035, Sep. 2020.
[6] L. Wu, X. Du, and X. Fu, “Security threats to mobile multimedia applications: Camera-based attacks on mobile phones,” IEEE Commun. Mag., vol. 52, no. 3, pp. 80–87, Mar. 2014.
[7] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, “Systematic classification of side-channel attacks: A case study for mobile devices,” IEEE Commun. Surveys Tuts., vol. 20, no. 1, pp. 465–488, 1st Quart., 2017.

[8] A. Maiti, M. Jadiwala, J. He, and I. Bilogrevic, “Side-channel inference attacks on mobile keyboards using smartwatches,” IEEE Trans. Mobile Comput., vol. 17, no. 9, pp. 2180–2194, Sep. 2018.

[9] S. Naual, A. Pandey, S. Gupta, G. Singal, V. Vinoth, and N. Nipo, “PIN inference attack: A threat to mobile security and smartphone-controlled robots,” IEEE Sensors J., early access, May 14, 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,” IEEE Trans. Mobile Comput., vol. 20, no. 2, pp. 337–351, Feb. 2021.

[11] Y. Zhang, M. Yang, G. Gu, and H. Chen, “Rethinking permission enforcement mechanism on mobile systems,” IEEE Inf. Forensics Security, vol. 11, no. 10, pp. 2227–2240, Oct. 2016.

[12] F. Roesner, “Designing application permission models that meet user expectations,” IEEE Secur. Privacy, vol. 15, no. 1, pp. 75–79, Jan. 2017.

[13] B. Rashidi, C. Fung, A. Nguyen, T. Vu, and E. Bertino, “Android user privacy preserving through crowdsourcing,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 3, pp. 773–787, Mar. 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,” IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3216–3225, Jul. 2018.

[15] J. Huang, Y. Xiong, W. Huang, C. Xu, and F. Miao, “SieveDroid: Intercepting undesirable private-data transmissions in Android applications,” IEEE Syst. J., vol. 14, no. 1, pp. 375–386, Mar. 2020.

[16] K. Gai, M. Qiu, and H. Zhao, “Privacy-preserving data encryption strategy for big data in mobile cloud computing,” IEEE Trans. Big Data, vol. 7, no. 4, pp. 678–688, Oct. 2021.

[17] H. Fu et al., “Towards automatic detection of nonfunctional sensitive transmissions in mobile applications,” IEEE Trans. Mobile Comput., vol. 20, no. 10, pp. 3066–3080, Oct. 2021, doi: 10.1109/TMC.2020.2992253.

[18] Y. Zhang et al., “Looking back! Using early versions of Android apps as attack vectors,” IEEE Trans. Dependable Secure Comput., vol. 18, no. 2, pp. 652–666, Mar. 2021.

[19] C. Ma, T. Wang, L. Shen, D. Liang, S. Chen, and D. You, “Communication-based attacks detection in Android applications,” Tsinghua Sci. Technol., vol. 24, no. 5, pp. 596–614, Oct. 2019.

[20] K. Chen, Y. Zhang, and P. Liu, “Leveraging information asymmetry to transform Android apps into self-defending code against repackaging attacks,” IEEE Trans. Mobile Comput., vol. 17, no. 8, pp. 1879–1893, Aug. 2018.

[21] L. Wu, X. Du, and J. Wu, “Effective defense schemes for phishing attacks on mobile computing platforms,” IEEE Trans. Veh. Technol., vol. 65, no. 8, pp. 6678–6691, Aug. 2016.

[22] H. Mustafa, W. Xu, A.-R. Sadeghi, and S. Schulz, “End-to-end detection of caller ID spoofing attacks,” IEEE Trans. Dependable Secure Comput., vol. 15, no. 3, pp. 423–436, May/ Jun. 2016.

[23] H. Ma, S. Li, D. Gao, D. Wu, Q. Jia, and C. Jia, “Active warden attack: On the ineffectiveness of Android app repackaging-proofing,” IEEE Trans. dependable secure comput., early access, Jul. 30, 2021, doi: 10.1109/TDSC.2021.3100877.

[24] H. Ma, S. Li, D. Gao, and C. Jia, “Secure repackaging-proofing framework for Android apps using collatz conjecture,” IEEE Trans. Dependable Secure Comput., early access, Jun. 22, 2021, doi: 10.1109/TDSC.2021.3091654.

**Misbah Shafi** (Student Member, IEEE) received the B.Tech degree from the Islamic University of Science and Technology, Jammu and Kashmir, India, and the M.Tech degree from Lovely Professional University, Punjab, India. She is currently pursuing the Ph.D. degree from the Shri Mata Vaishno Devi University, Jammu and Kashmir.

Her research interests include network security and wireless communication. She is currently doing her research on security issues of 5G, 5G NR, and 6G.

**Rakesh Kumar Jha** (Senior Member, IEEE) has worked as an Associate Professor at Shri Mata Vaishno Devi (SMVD) University, Katra, Jammu and Kashmir, India. He is currently an Associate Professor with the Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, India. He is among the top 2% researchers in the world. He has published more than 41 Science Citation Index journal papers, including many IEEE TRANSACTIONS, IEEE journals, and more than 25 international conference papers. His areas of interest are wireless communication, optical fiber communication, computer networks, and security issues. His one concept related to the router of wireless communication was accepted by ITU in 2010.

Prof. Jha is a member of ACM and CSI, with many patents and more than 1941 citations on his credit. He is a Senior Member of GISFI, SIAM, the International Association of Engineers (IAENG), and the Advance Computing and Communication Society (ACCS). He received the Young Scientist Author Award by ITU in December 2010. He received an APAN Fellowship in 2011, 2012, 2017, and 2018 and a Student Travel Grant from COMSNET 2012.

**Sanjeev Jain** (Senior Member, IEEE) is currently working as a Vice-Chancellor of the Central University of Jammu, Jammu and Kashmir, India. He has worked as the Director of the Indian Institute of Information Technology, Design and Manufacturing Jabalpur, India. He has also worked as a Vice-Chancellor at Shri Mata Vaishno Devi (SMVD) University, Katra. He has served as the Director of the Madhav Institute of Technology and Science (MITS), Gwalior. Besides teaching, he has the credit of making a significant contribution to research and development in the area of image processing and mobile ad hoc networks. His work on digital watermarking for image authentication is highly valued in the research field.