Journal of Ambient Intelligence and Humanized Computing (2023) 14:4785–4794
https://doi.org/10.1007/s12652-023-04535-7

ORIGINAL RESEARCH

Circumventing Google Play vetting policies: a stealthy cyberattack that uses incremental updates to breach privacy

Zia Muhammad[1,2] · Faisal Amjad[3] · Zafar Iqbal[2] · Abdul Rehman Javed[2,5] · Thippa Reddy Gadekallu[4,5]

Received: 31 March 2022 / Accepted: 12 January 2023 / Published online: 28 January 2023

Abstract

Today digital technologies are evolving to accommodate small businesses and young entrepreneurs by reducing their time-to-market while encouraging rapid innovation in mobile, Extended Reality (XR), Internet of Things (IoT), cloud, and edge devices. The leading operating system Android typically takes one to a few days to perform application vetting and go to production by leveraging code analysis technologies in their Play Protect anti-malware program. However, developers with malicious intent are looking to circumvent this detection mechanism by exploiting Google's relatively lenient trust policies that allow for package distribution and feature updates. This paper develops a proof-of-concept malware that exploits customers' trust and Google's policies to circumvent popular voice search applications. Our results show that attackers can initially circumvent Play Protect by uploading benign applications to build trust and then add malicious feature updates incrementally to distribute highly intrusive malware into user systems. This malware can scan and collect private user data from the device and exfiltrate it to the command-and-control server. The contributions are three-fold. (1) A proof-of-concept stealthy malware and publishing mechanism has developed that highlights the relative ease with which Google Play Protect policies may be subverted. (2) a comprehensive evaluation has been performed using major publicly available anti-malware solutions. (3) Recommendations and policies have been suggested to prevent this attack and ensure users' privacy concerns (IMUTA is a novel attack in which malicious functionality is slowly added to a benign application through updates. This attack evades malware detection tools and exploits user trust. The attack can be launched against any application distribution platform like the Play Store).

Keywords Mobile security · Android security · Google play · Play store · Play protect · Android malware · Innovation and growth · Security policies

1 Introduction

To meet current technological advancements, many organizations are working to encourage the continuous expansion of entrepreneurs, young businesses, and small enterprises Rehman et al. (2022); Javed et al. (2020). It could take a lot of time and effort to investigate each one to determine its potential value. However, Google is providing growth opportunities with simple-to-use tools for businesses and organizations to manage their needs. It provides Google Play, a general application distribution mechanism that provides an opportunity to millions of entrepreneurs, young developers,

✉ Thippa Reddy Gadekallu
thippareddy.g@vit.ac.in

Zia Muhammad
zia.muhammad@ndsu.edu; zia.muhammad@mail.au.edu.pk

Faisal Amjad
faisal@nust.edu.pk

Zafar Iqbal
zafar.iqbal@mail.au.edu.pk

Abdul Rehman Javed
abdulrehman.cs@au.edu.pk

[1] Department of Computer Science, North Dakota State University, Fargo, USA

[2] Department of Cyber Security, PAF Complex, E-9, Air University, Islamabad, Pakistan

[3] Department of Information Security, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[4] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India

[5] Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon

and global institutions to submit, publish, and market their mobile applications and games. Their publication mechanism is relatively simple and takes around one to a few days to market an application. This is a good opportunity to cash and earn handsome money using Google AdMob's in-app advertising facility. Their user and developer-friendly application publication policies provide ease of understanding, and it has become relatively simple for a new developer to develop and upload an application to Google play. Moreover, Google builds a trust factor with old publishers and provides efficiency in production time. However, there can be a negative impact, as developers with malicious intent want to circumvent this detection mechanism by exploiting Google's relatively lenient trust policies. Sometimes users' privacy can be affected by these linear policies. Nowadays, there is a misconception that one can be safe if the applications are downloaded only from Google Play, which is not the case anymore. Instead, If these application stores are not secured, they can lead to global attacks with various objectives such as financial gain, political gain, sabotage, and Personal Identifiable Information (PII) leaks.

There is undoubtedly no end to the significance of Android-based devices in our daily lives Javed et al. (2022). Android is the sole ownership of Google and is used in home security systems, Smartwatches, TV boxes, car navigation, minicomputers, and notably smartphones. In the past few years, the use of Android mobile phones has risen exponentially. As per 2022 statistics, Report (2022), Google claims more than 2.5 billion active devices worldwide. Android is Linux-based, and its open-source nature makes it a popular platform among all kinds of cellphone manufacturers. This global distribution of the Android OS makes it a superlative target for cybercriminals Saracino et al. (2018); Rasool et al. (2021); Imtiaz et al. (2021); Javed et al. (2021). Therefore, various types of malware are developed to target Android devices. They get installed on Android devices through various means and steal users' Personal Identifiable Information (PII) like device details, contacts, messages, call logs, user location, images, and linked accounts Narayanan et al. (2017).

Android edge devices provide Google Play to download and update applications with built-in malware protection mechanisms. The research explicitly targets Android-based smartphone categories. Multiple Android application stores are publicly available and provide both free and paid applications, but Google Play is popular among all Zhao et al. (2022); Viennot et al. (2014); Kumar et al. (2021). The platform is one of the first mobile stores that opened in 2008. It is considered the most widespread, providing more than 3 million free and paid applications as of Jan 2022 buildfire (2022). Google has sole ownership over the Play Store and announced it as a default application distribution and update platform for Android-based devices. To ensure the security of end-users, Google has introduced several security procedures. It has developed a special skill set to detect any malicious application uploaded to its distribution servers Lee (2019). In the past, Google used Bouncer to protect Play Store security from malicious applications. It classified malware as spyware, trojan, adware, and backdoor. Due to a security Infrastructure update held in 2017, Google Bouncer was replaced with *Google Play Protect*, and it was embedded in all Android devices to identify potentially harmful applications (PHA) Mirza et al. (2021). Play Protect is a multi-tiered malware detection system that scans every application uploaded on the Play Store and performs routine scans on a mobile device to find and remove suspicious applications Ahmed et al. (2021). Moreover, it is also responsible for performing heuristic malware analysis, including but not limited to monitoring network activity, malicious links, background services, and insecure data disclosure Renjith and Aji (2022); Liyanage et al. (2017); Ranaweera et al. (2019).

As Android is purely based on Security Enhanced Linux (*SELinux*) McCarty (2005). Policy enforcement mechanisms ensure security at the root level by providing OS-level controls, access vector cache, object-based security, SELinux enforcement mode, mandatory access controls, user ID, and process ID. Moreover, *Google security measures* include Play Protect, Google safety checks, permission declaration, play console, generate privacy/ security alerts, identity misrepresenting application information, and release updates for device security Muhammad et al. (2021); Fatima et al. (2021); Usman et al. (2021). Finally, *OEM security measures* include security patches, update management, Knox security platform, antivirus application, run time application check, secure vaults, and multi-layer security and tracking systems. Additionally, to maintain mobile security round-the-globe, Google uses SafetyNet Lu et al. (2017); Alazab and Tang (2019) for intrusion detection, privacy preservation, and identification of new security threats Roy et al. (2022); Srivastava et al. (2022). Furthermore, Compatibility Test Suite (CTS) has been used to check a device's compatibility with the application server.

In this paper, Google Play vetting policies, user security, and malware detection capabilities have been audited and evaluated. For this, a custom malware application has been developed, and the Incremental Malicious Update Attack (IMUTA) technique has been demonstrated. This experiment is performed on the Android Play Store over one year and two months (61 weeks/ 427 days). Our experiment started on October 31, 2020, and ended on January 1, 2022.

1.1 Contribution

1. Study of the Android ecosystem, taxonomy, security posture, permission model, versions, architecture,

Table 1 Comparison of recent work with the scope of this paper

Author	Publication Year	Android Basics	Security Posture	Attack Simulation	Results and Evaluation	Threat Defense
Alex, and Jerome Allix et al. (2014)	2014	✓	✓	×	×	✓
F. Mercaldo Mercaldo et al. (2016)	2016	✓	×	×	✓	×
S. Hutchinson Hutchinson et al. (2019)	2019	×	×	✓	✓	×
K. Tian Tian et al. (2020)	2020	×	✓	×	✓	✓
Shalaginov Shalaginov (2021)	2021	✓	✓	×	×	✓
M. Isabel Montano et al. (2021)	2021	✓	×	×	✓	×
Cao. Michael Cao (2022)	2022	×	✓	×	×	✓
Our Paper	2022	✓	✓	✓	✓	✓

Android Application Packages (APK), and permission model.

2. Development and deployment of *Voice Search* Application on Google Play Store. A benign application that is converted to stealthy malware by launching multiple updates collects and exfiltrates desired data to the intended destination.
3. A comprehensive evaluation has been performed using major publicly available anti-malware solutions to measure the effectiveness and accuracy of the proposed model.
4. Finally, recommendations and policies have been suggested to prevent this attack and ensure users' privacy concerns.

Organization of the paper: The paper is categorized into the following sections: Sect. 2 provides a background fundamentals and related work presented in recent years. Section 3 explains the core methodology and application baseline to bypass the security of Google Play Protection. Section 4 explains the overall design and architecture of the Voice Search application. Section 5 provides effectiveness, accuracy, and details of the experiment timeline. Section 6 provide detailed discussion and improvement suggestions. Lastly, Sect. 7 provides conclusions and future directions.

2 Literature review

Due to the novelty of our work, we found comparatively less literature in our domain. However, several authors have nominated different aspects of Android device security, like the creation, propagation, and detection of malware among Android devices. Moreover, details of the Play Store and Play protect have been discussed in this section.

2.1 Google play store and play protect

Play Store is an official application of Google that comes pre-installed on certified-Android devices. It is a place for people to go and find their required apps, movies, games, books, Tv shows, and much more Karunanayake et al. (2022). Play Store provides its services in more than 190 countries. In parallel, it provides millions of applications in multiple categories, including but not limited to communication, lifestyle, business, and productivity. There are more than 53 categories freely available for users to choose and download desired applications.

Play Protect is a built-in security solution that protects Android devices from Potentially Harmful Applications (PHA). It is one of the most widely spread threat protection solutions, securing over 2.5 billion Android devices. It follows a layered security model to build a strong defense against malware. The core protection areas of Play Protect include Android OS, and applications Sharma et al. (2022). It is an artificial intelligence-based system continuously evolving and strengthening its defense against malicious applications. The critical functionality is as follows:

- It runs a safety check against every application downloaded from the Play Store.
- It removes applications from the Play Store that violate users' privacy and confidentiality and exposes users' data.
- It helps to keep the device secure by scanning it at specific time intervals.

2.2 Android attacks and defence mechanisms

This section provides a detailed review of various past efforts in the chosen domain. It includes state-of-the-art techniques proposed in the last decade. Moreover, a logical categorization and comparative analysis have been added in Table. 1. Alex et al. Allix et al. (2014) described the working principle of Android malware. The researchers used forensic tools to

visualize the behavior of malware. This article envisioned multiple aspects, such as malware propagation, working principle, and activity flow, were envisioned. However, no research has been conducted on propagation through third-party stores. In 2016, Mercaldo et al. Mercaldo et al. (2016) introduced a detection model against newly launched application update attacks in the mobile malware landscape. They countered some traditional malware propagation techniques. Due to the rapidly changing threat landscape, the proposed model does not seems useful for modern threats. Moreover, Hutchinson et al. Hutchinson et al. (2019) developed a spyware application and published it on AppStore. The researcher used demystified permission model vulnerability, but this vulnerability was patched with a substantial delay.

Tian et al. Tian et al. (2020) described a functional model to detect repackaged malware. This research briefly explains how a code is injected into an existing application by exploiting the actual behavior and making it undetectable. This malware is hard to detect, and their proposed technique is a valuable contribution. Like malware detection, malware classification is considered vital to see the efficiency of malware detection capabilities of tools. The authors Shalaginov (2021) researched existing Android malware classification and categorization techniques. They proposed a model based on modus operandi and existing malware vendor reports to classify the state of the malware. Isabel et al. Montano et al. (2021) claimed that Google Play Protect employed multiple types of time complexity algorithms for spam detection. It classifies applications into different categories based on their functionality and performs routine checkups to detect suspicious activities on all installed applications. Furthermore, a rule-based filtering mechanism is used to enhance the throughput of the detection algorithm.

Finally, Michael et al. Cao (2022) write that Google Play Store is vulnerable to invasive malware threats. The article thoroughly studies the Play Store security mechanism and malware samples in January 2016 and July 2021. The author worked on identifying malware families using control and data-flow graphs. The work is focused on defensive measures and does not possess any attack and simulation to breach the application store.

These works are comprehensive efforts to ensure the Android malware detection and propagation processes. Although discussed works are excellent motivation in research, none of these studies evaluate Google Play Protect services against custom-designed malware updates. In contrast, our research analyzes the Google Play Protect detection mechanism and demonstrates a malicious bypass. Details of the experiments are added in the subsequent sections.

Moreover, they can counter the latest malware threats to some extent. These tools include RoughDroid, DroidDector, MADAM, MalDozer, SeqDroid, DL-Droid, ProDroid, and MDTA. These tools used machine learning, deep learning, and deep belief networks to detect and classify malware.

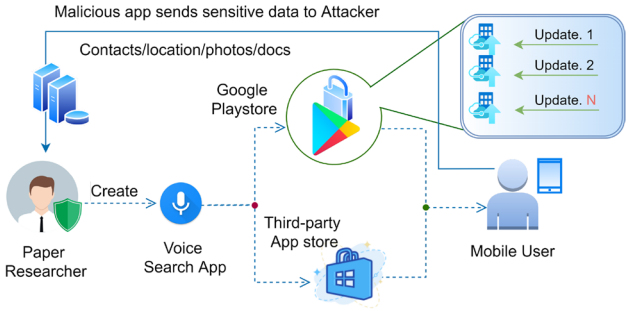

Fig. 1 The proposed methodology that employs an incremental malicious update mechanism to breach Google Play Protect Security

3 Methodology

According to Google's Android security report, [37] released in 2018, applications installed from the Google Play Store are eight times more secure than applications installed from other stores. It is due to the in-depth analysis of the application performed by Google. When an application is submitted to the Google Console, Google starts the review process. This review procedure generally takes one to many days to approve an application. The application is made public when an application qualifies against the developer distribution agreement, and no policy violation is found.

Our methodology aims to develop an application to evade Google Play Protect security checks. During the literature review, it was found that Google Play Protect can detect and block an application that tries to install Over the Air (OTA) updates that the application directly accesses from a third-party server. However, there is a possibility to evade Play Protect security mechanisms without the involvement of any third-party server. The working diagram of our proposed methodology is added in Fig. 1. The diagram highlights an application created and uploaded to the Play Store. Afterward, multiple updates are released.

4 Voice search—design and architecture

Details of the entire process have been added in this section. The key steps involved in our research are as follows:

1. First of all, a benign application name "Voice Search V1.1" is developed and uploaded to the Google Play Store. It allows users to perform various actions through voice commands, such as calling someone or finding the latest news/ weather. This application version was submitted to Google Console on *October 31, 2020*. Google took seven days to review and accept the application as legitimate on *November 6, 2020*. Since then, the application has been publicly available.

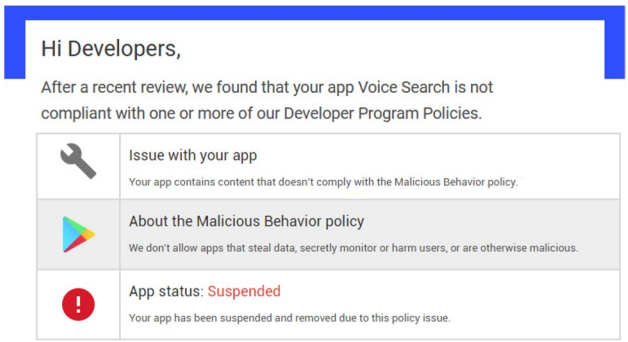

Fig. 2 Google detected and blocked malicious application that was directly uploaded to Google play store

2. The second version of the Voice Search application was developed and named V1.2. In this version, malicious functionality has been introduced that access and uses analytics, event logs, activity tracking, demographics, and user location. This update was submitted to Google Console on *November 16, 2020* and was accepted on *November 17, 2020*. Google took just one day to review our update and made it public.
3. Subsequently, a third version, V1.3, was developed. This was an entirely malicious update. This version was capable of creating a reverse connection on Firebase storage Stonehem (2016) to store data against each user's entity. It was capable of exploiting Android, and it started collecting device contacts, version, API level, manufacture, and model details. The user's data was collected when the user opened the application and performed a voice action. This version was submitted to Google Console on *December 26, 2020* and was accepted on *December 27, 2020*. Google took one day to review our updates thoroughly, and this malicious version was public on the Play Store.

The above experiment shows that Google conducts an in-depth analysis that takes as much time as possible and reviews an application in depth when it is initially published on the Play Store. Then, it takes less time to review when an update is made. It is difficult to upload an utterly malicious application, but the same can be done in multiple updates. At the beginning of the experiment, an utterly malicious application was directly uploaded, but Google detected it, removed it, and sent a suspension letter. Notice of suspension can be seen in Fig. 2. On the other hand, the same malicious application is uploaded to the Google Play Store through multiple updates.

5 Evaluation

The section is divided into three sub-sections that provide details of the proposed model's effectiveness, accuracy, and experiment timeline. (1) Effectiveness covers experiment novelty and detectability against malware detection solutions. (2) Accuracy in user data acquisition, collection, and storage. (3) Experiment timeline provides timestamps and the number of days for each application publication phase.

Table 2 Test results of developed malware against Android Antivirus/ Anti-malware tools

No	Tool Name	Tool Version	Android devices	Detection
1	Play Protect	27.9	Android 11	×
2	Avast	6.35	Android 11	×
3	AVG	6.35	Android 10	×
4	Avira	7.4	Pie 9.0	×
5	Bitdefender	3.3	Android 11	×
6	Kaspersky	11.6	Oreo 8.0	×
7	McAfee	5.9	Oreo 8.0	×
8	Norton	5.1	Nougat 7.0	×
9	TOTALAV	2.0	Pie 9.0	×
10	Mobile Security	12.1	Oreo 8.0	×

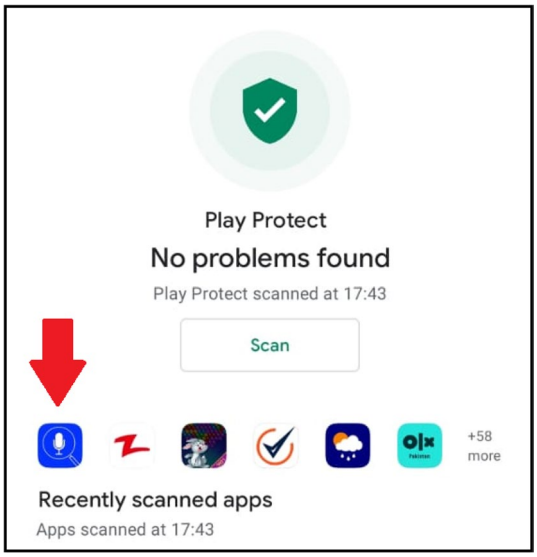

Fig. 3 Google play protect scanning process: Google scanned developed application and considered it safe although it was malicious

5.1 Effectiveness

To measure the effectiveness of our approach, the application was downloaded on multiple Android devices to test against Google Play Protect and other antivirus solutions, but none was able to detect our malicious application. Results of antivirus solutions have been added in Table. 2. Moreover, Fig. 3 shows a device screenshot in which Google Play Protect scanned all applications, including our malicious application, but our application is classified as benign. Although Play Protect performs a routine analysis

on installed applications in the device and checks against the potentially harmful activity, it could still not detect this kind of attack. None of the antivirus solutions were able to detect it.

The proposed methodology provided an effective way to propagate a malicious application through Google Play Store. There is a possibility that an attacker may use a similar method to create and publish an application to exploit users' data privacy. If this kind of attack is launched against a specific audience with an intention, it will be widespread and hard to stop.

5.2 Accuracy

We measured the accuracy of the proposed solution based on results acquired during the entire experimentation process. This section provides details of users' data collected throughout our experiment. The collected data is organized in a specific order for better understanding. All the collected data is stored in *Firebase*[1] servers. The data collection phase is triggered when the application user performs a voice actor for the first time. Subsequently, the voice query is entertained, and required data is collected in parallel to store on Firebase. The collected data is divided into two sections, i.e., *Data Stored in Firebase* and *User analytic*.

1. A new user entity is created in Firebase with a unique cryptographic key.
2. Device data is uploaded against the user.
3. Afterwards, the device is marked successful in avoiding feature repetitions. A quick overview of discussed functionality is provided in Algorithm. 1.

Algorithm 1: Application data collection phase

```
function voiceAction (U_v r, N_u ser);
Input  : User Voice Recording (U_v r) and Status of user (N_u ser)
Output: Data backup against Firebase Instance
if N_u ser = 1 then
    performVoiceAction(U_v r);
    x=getFirebaseInstance();
    y=getDeviceDetails();
    z=getUseranalytics();
    firebase.backup(x, y, z);
else
    performVoiceAction(U_v r);
    /*User voice query will be entertained*/
end
```

5.2.1 Data stored in firebase

This section provides the highlights of data stored in a Firebase server. Figure 4 shows data stored in Firebase against every user. This data is stored in a particular order, and a quick overview is as follows:

- **MPZa7B5ywvX2lP-Yx1w:** This entity contains a Firebase unique key for individual devices.
- **Manufacturer:** It contains the name of the mobile company that is the manufacturer of the device.
- **Android version:** This entity contains Android version numbers like Android 10, 9, and 8.
- **API level:** It contains an integer value of the used API.
- **Info:** This entity contains a string that has device contacts locally stored in the system.
- **Demographics:** This shows the overall statistical view for the gender/ age group of the audience of the application.
- **Location:** These are statistics for countries/ regions in which the application is installed.
- **Affinity Audience:** These statistics give a detailed report about application users based on their interests, lifestyle, habits, passion, and online activities.

5.2.2 User analytic

This section presents a graphical representation of all the users who installed the application. This can be seen that most of the audience is from India and Russia, followed by other countries.

Figure 5 contains a graphical representation of the targeted audience based on their ages. All users are classified into age groups and genders. The minimum age group

[1] **Firebase**®. It is a cloud storage platform launched by Google and is considered the most trusted platform for storage, analytics, and backend services for mobile and web applications.

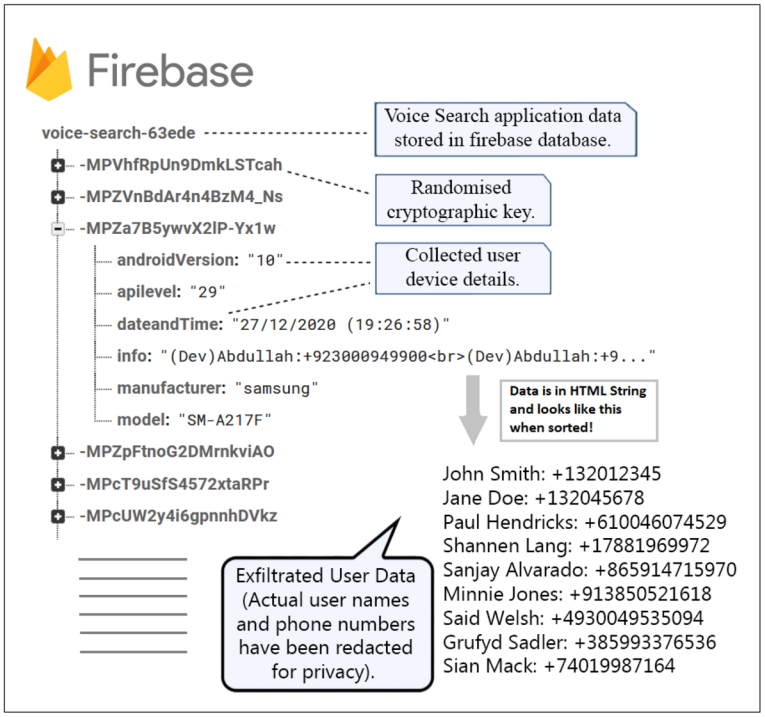

Fig. 4 Graphical representation of Firebase data storage hierarchy: the snapshot visualizes collected user data stored in Firebase cloud service

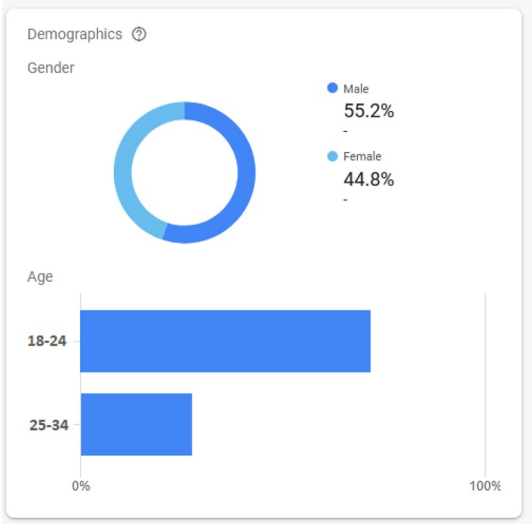

Fig. 5 Graphical representation of Users' age group/ gender collected by voice search

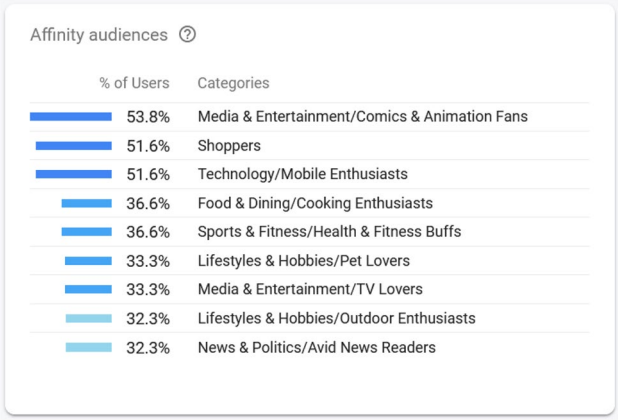

Fig. 6 Graphical presentation of users' data relevant to their interests, lifestyle, habits, passion, and online activities collected by voice search application

targeted in our application is 18 years. Figure 6 gives a brief overview of our affinity audiences. It categorizes users into their respective categories based on past activities, interests, and hobbies. The Firebase analytic algorithm keeps track of users' daily activities and assigns an affinity group based on their recent behavior.

Attackers can attach such malicious pieces of code with the popular Android application to steal data. Further, this kind of application can be propagated to the desired audience using an Android messaging service. After installing the application and collecting desired data from devices without getting noticed by Play Protect, it can be concluded that such attacks are hard to detect.

5.3 Experiment timeline

In the initial experiment, three versions of the Android application were created and published on the Google Play Store. Table. 3 contains application publication details against their version number. The table has three columns application version, processing time that Google took for evaluation before

Table 3 Timeline of an experiment for benign and malic

Version	Processing time	Functionality
V 1.1	7 Days *Submission:* 31-10-2020 *Acceptance:*	- Benign App. - App permissions. - 12,782 devices. - 151 countries
V 1.2	1 Day *Submission*: 16-11-2020 *Acceptance:* 17-11-2020	- Added analytics. - Event logs. - Demographics. - Userbase Location. - Affinity Audience
V 1.3	1 Day *Submission:* 26-12-2020 *Acceptance:* 27-12-2020	- Permissions exploit. - Firebase data backup. - Contacts backup. - Device Details. - Data is back up on voice action
V 1.4	1 Days *Submission:* 1-1-2022 *Acceptance:* 1-1-2022	- Benign App. - Legitimate functionality. - 12,782 devices. - 151 countries

making a version live on Google Play Store, and functionality. After this successful experiment that continued for one year and two months, the application was updated to V1.4. A fully benign and legitimate update without any malicious functionality. Version 1.4 was updated and accepted by Google on January 1, 2022.

6 Discussion and recommendation

The main objective of this research is to assess the security of Google Play Protect. Attackers may evade the application screening process and steal user data. This research indicates that installed antivirus and anti-malware solutions cannot prevent these attacks. While uploading a new update, Google compares the application package name, ID, incremented version, and Keystore with an earlier version of the application. However, additional features Like added functionality, code enrichments, and application content is not compared with an earlier version. So, it is possible to update an existing version with a different application ensuring that it has the same package name, ID, incremented version, and Keystore. To defend against this kind of attack, one can enrich vetting policies. Suggested policy-based recommendations and process improvement are as follows:

1. Acquire detailed information on the updated application module from a developer. Afterward, calculate the similarity index based on the code similarity of the earlier version with the new variant. This similarity index will help to spot the difference; a higher similarity index means an application is slightly updated, while a lower similarity index means more amendments have been performed.
2. This would be effective to merge new code in the previous version instead of replacing an application entirely with an update. The same functionality is used in Version Control Systems (VCS), and new code is merged with earlier versions.
3. Table. 3 show that the Google Play Store had taken seven days to scan the application when it was first published but less than a day to scan when the application was updated. There is a possibility that an initial application is evaluated more critically than application updates, so updates got published quickly.
4. Critical analysis of application updates is required to protect Android devices from such attacks. The analysis must go through the same process as the evaluation of the initial application performed; this may include maliciousness, hidden intents, requested permissions, provided functionality, and comparing the code of published versions of the application and its updates.

7 Conclusion and future extensions

This is assumed that users can download safe and authentic applications from Google play. People trust Google, but its flexible application vetting policy may endanger common user data security. However, Google performs an in-depth analysis on submitted applications using a built-in malware protection mechanism called Google Play Protect that prevents malicious applications from being installed, but there is still a gap for certain advancements. The research is an extensive effort in the mobile security paradigm that invites security researchers, analysts, and software developers to investigate and combat these security breaches. Moreover, it makes the general consumer aware that publicly available application stores cannot be blindly trusted. Finally, it invites the researcher to analyze the security mechanisms of other application distribution platforms available in various operating systems such as Windows, iOS, Mac, IoT, cloud, and edge devices.

Funding This research is funded by Sheila and Robert Challey Institute for Global Innovation and Growth, North Dakota State University (NDSU), USA.

Data Availability Data sources are highlighted in the paper.

Declarations

Conflict of interest/Conflict of interest The authors share no conflict of interests.

References

Ahmed W, Rasool A, Javed AR, Kumar N, Gadekallu TR, Jalil Z, Kryvinska N (2021) Security in next generation mobile payment systems: a comprehensive survey. IEEE Access

Alazab M, Tang M (2019) Deep learning applications for cyber security. Springer, Cham

Allix K, Jerome Q, Bissyande TF, Klein J, State R, Traon YL (2014) A Forensic Analysis of Android Malware. In: 38th Annual Computer Software and Applications Conference, IEEE, pp 384–393, 10.1109/COMPSAC.2014.61, http://ieeexplore.ieee.org/document/6899240/. Accessed 22 July 2022

Buildfire (2022) Ultimate mobile app stores list. https://www.android.com/intl/en_us/intl/en_uk/play-protect/, last checked on Jan 7, 2022

Cao M (2022) Understanding the characteristics of invasive malware from the google play store. PhD thesis, University of British Columbia

Fatima M, Abbas H, Yaqoob T, Shafqat N, Ahmad Z, Zeeshan R, Muhammad Z, Rana T, Mussiraliyeva S (2021) A survey on common criteria (cc) evaluating schemes for security assessment of it products. PeerJ Comput Sci 7:e701

Google (2018) Android Security and Privacy 2018 Year In Review. https://source.android.com/security/reports. Report Dec, 2020

Hutchinson S, Zhou B, Karabiyik U (2019) Are we really protected? An investigation into the play protect service. In: 2019 IEEE International Conference on Big Data (Big Data), pp 4997–5004, 10.1109/BigData47090.2019.9006100

Imtiaz SI, Imtiaz SI, ur Rehman S, Javed AR, Jalil Z, Liu X, Alnumay WS (2021) Deepamd: detection and identification of android malware using high-efficient deep artificial neural network. Future Gen Comput Syst 115:844–856

Javed AR, Beg MO, Asim M, Baker T, Al-Bayatti AH (2020) Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes. J Ambient Intell Human Comput. pp 1–14

Javed AR, Rehman SU, Khan MU, Alazab M, Khan HU (2021) Betalogger: smartphone sensor-based side-channel attack detection and text inference using language modeling and dense multilayer neural network. Trans Asian Low-Resour Lang Inf Process 20(5):1–17

Javed AR, Shahzad F, ur Rehman S, Zikria YB, Razzak I, Jalil Z, Xu G (2022) Future smart cities requirements, emerging technologies, applications, challenges, and future aspects. Cities 129:103794

Karunanayake N, Rajasegaran J, Gunathillake A, Seneviratne S, Jourjon G (2022) A multi-modal neural embeddings approach for detecting mobile counterfeit apps: A case study on google play store. IEEE Trans Mob Comput 21(1):16–30. https://doi.org/10.1109/TMC.2020.3007260

Kumar A, Sharma A, Bharti V, Singh AK, Singh SK, Saxena S (2021) Mobihisnet: a lightweight cnn in mobile edge computing for histopathological image classification. IEEE Internet Things J 8(24):17778–17789

Lee W (2019) SeqDroid: obfuscated android malware detection using stacked convolutional. In: deep learning applications for cyber security. Springer International Publishing, Cham, pp 197–210, https://doi.org/10.1007/978-3-030-13057-2_9, http://link.springer.com/10.1007/978-3-030-13057-2_9

Liyanage M, Ahmed I, Okwuibe J, Ylianttila M, Kabir H, Santos JL, Kantola R, Perez OL, Itzazelaia MU, De Oca EM (2017) Enhancing security of software defined mobile networks. IEEE Access 5:9422–9438

Lu J, Issaranon T, Forsyth D (2017) Safetynet: Detecting and rejecting adversarial examples robustly. In: Proceedings of the IEEE international conference on computer vision. pp 446–454

McCarty B (2005) SELinux. O'Reilly Japan

Mercaldo F, Nardone V, Santone A, Visaggio CA (2016) Download malware? no, thanks: How formal methods can block update attacks. In: Proceedings of the 4th FME Workshop on Formal Methods in Software Engineering, Association for Computing Machinery, New York, NY, USA, FormaliSE '16, p 22-28, https://doi.org/10.1145/2897667.2897673

Mirza S, Abbas H, Shahid WB, Shafqat N, Fugini M, Iqbal Z, Muhammad Z (2021) A malware evasion technique for auditing android anti-malware solutions. In: 2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE. pp 125–130

Montano IH, de la Torre Díez I, López-Izquierdo R, Villamor MAC, Martín-Rodríguez F (2021) Mobile triage applications: a systematic review in literature and play store. J Med Syst 45(9):1–11

Muhammad Z, Amjad MF, Abbas H, Iqbal Z, Azhar A, Yasin A, Iesar H (2021) A systematic evaluation of android anti-malware tools for detection of contemporary malware. In: 2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC), IEEE. pp 117–124

Narayanan A, Chandramohan M, Chen L, Liu Y (2017) Context-aware, adaptive, and scalable android malware detection through online learning. IEEE Trans Emerg Topics Comput Intell. 1(3):157–175. https://doi.org/10.1109/TETCI.2017.2699220

Ranaweera P, Jurcut AD, Liyanage M (2019) Realizing multi-access edge computing feasibility: security perspective. In: 2019 IEEE Conference on Standards for Communications and Networking (CSCN), IEEE. pp 1–7

Rasool A, Javed AR, Jalil Z (2021) Sha-amd: sample-efficient hypertuned approach for detection and identification of android malware family and category. Int J Ad Hoc Ubiquitous Comput 38(1–3):172–183

Rehman A, Razzak I, Xu G (2022) Federated learning for privacy preservation of healthcare data from smartphone-based side-channel attacks. IEEE J Biomed Health Inform

Renjith G, Aji S (2022) Unveiling the security vulnerabilities in android operating system. In: Proceedings of Second International Conference on Sustainable Expert Systems. Springer, Cham. pp 89–100

Report AS (2022) Google play protects 2.5 billion active devices. https://www.android.com/intl/en_us/intl/en_uk/play-protect/, last checked on Jan 4, 2022

Roy AK, Nath K, Srivastava G, Gadekallu TR, Lin JCW (2022) Privacy preserving multi-party key exchange protocol for wireless mesh networks. Sensors 22(5):1958

Saracino A, Sgandurra D, Dini G, Martinelli F (2018) MADAM. IEEE Trans Depend Secure Comput. 15(1):83–97. https://doi.org/10.1109/TDSC.2016.2536605

Shalaginov A (2021) Review of the malware categorization in the era of changing landscape. Malware Analysis Using Artificial Intelligence. Springer, Cham

Sharma S, Khanna K, Ahlawat P (2022) Survey for detection and analysis of android malware (s) through artificial intelligence techniques. Cyber security and digital forensics. Springer, Cham, pp 321–337

Srivastava G, Jhaveri RH, Bhattacharya S, Pandya S, Maddikunta PKR, Yenduri G, Hall JG, Alazab M, Gadekallu TR, et al. (2022) Xai for cybersecurity: State of the art, challenges, open issues and future directions. arXiv preprint arXiv:2206.03585

Stonehem B (2016) Google android firebase: learning the basics, vol 1. First Rank Publishing

Tian K, Yao D, Ryder BG, Tan G, Peng G (2020) Detection of repackaged android malware with code-heterogeneity. IEEE Trans Depend Secure Comput 17(01):64–77. https://doi.org/10.1109/TDSC.2017.2745575

Usman N, Usman S, Khan F, Jan MA, Sajid A, Alazab M, Watters P (2021) Intelligent dynamic malware detection using machine

learning in ip reputation for forensics data analytics. Future Gen Comput Syst 118:124–141

Viennot N, Garcia E, Nieh J (2014) A measurement study of google play. In: The 2014 ACM international conference on Measurement and modeling of computer systems - SIGMETRICS '14, ACM Press, Austin, Texas, USA, pp 221–233, https://doi.org/10.1145/2591971.2592003, http://dl.acm.org/citation.cfm?doid=2591971.2592003

Zhao J, Cao B, Liu X, Yang P, Singh AK, Lv Z (2022) Multiobjective multiple mobile sink scheduling via evolutionary fuzzy rough neural network for wireless sensor networks. IEEE Trans Fuzzy Syst