information

MDPI

Article

Assessing the Security and Privacy of Android Official ID Wallet Apps

Vasileios Kouliaridis [1,*,†], Georgios Karopoulos [1,†] and Georgios Kambourakis [2,†]

[1] European Commission, Joint Research Centre (JRC), 21027 Ispra, Italy; georgios.karopoulos@ec.europa.eu
[2] Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovasi, Greece; gkamb@aegean.gr
* Correspondence: vasileios.kouliaridis@ec.europa.eu
† These authors contributed equally to this work.

Abstract: With the increasing use of smartphones for a wide variety of online services, states and countries are issuing official applications to store government-issued documents that can be used for identification (e.g., electronic identity cards), health (e.g., vaccination certificates), and transport (e.g., driver's licenses). However, the privacy and security risks associated with the storage of sensitive personal information on such apps are a major concern. This work presents a thorough analysis of official Android wallet apps, focusing mainly on apps used to store identification documents and/or driver's licenses. Specifically, we examine the security and privacy level of such apps using three analysis tools and discuss the key findings and the risks involved. We additionally explore Android app security best practices and various security measures that can be employed to mitigate these risks, such as updating deprecated components and libraries. Altogether, our findings demonstrate that, while there are various security measures available, there is still a need for more comprehensive solutions to address the privacy and security risks associated with the use of Android wallet apps.

Keywords: Android privacy; Android security; wallet apps

Citation: Kouliaridis, V.; Karopoulos, G.; Kambourakis, G. Assessing the Security and Privacy of Android Official ID Wallet Apps. *Information* **2023**, *14*, 457. https://doi.org/10.3390/info14080457

Academic Editors: Jose de Vasconcelos, Hugo Barbosa and Carla Cordeiro

Received: 24 July 2023
Revised: 10 August 2023
Accepted: 11 August 2023
Published: 13 August 2023

1. Introduction

The increasing reliance on mobile devices for accessing online services has led to the development of various wallet applications (apps), which allow citizens to upload and store government issued documents, such as vaccination certificates, identity documents (IDs), and driver's licenses (DLs). The electronic IDs and DLs stored in a wallet app contain the same information as their physical counterparts, i.e., personal information such as name, date of birth, and photo, as well as a unique identifier, document issue, and expiration date. Depending on the state or country of issue, these electronic copies may be used for various purposes, including accessing government services, opening bank accounts, conducting online transactions, identification in public services, and police inspections. The European Commission has proposed a European Digital Identity using a digital wallet [1]; more recently, the launch of EU-wide digital driving licenses [2] was also announced. In addition, the US Transportation Security Administration (TSA) currently accepts some mobile IDs and DLs in a number of US airports [3]. With regard to the end user, it is foreseen that, globally, one in two people will use a mobile wallet by 2025 [4], increasing the number of mobile wallets in use from 2.8 billion at the end of 2020 to 4.8 billion by the end of 2025.

While holding all identification documents in one place may seem convenient, it also raises serious concerns regarding the privacy and security offered by these apps to users and their sensitive personal information, as shown in relevant research [5–10]. The potential risks associated with the storage of ID copies on Android apps are numerous. For instance, the mobile device can be lost or stolen, or the app can be compromised by malware or other more direct attacks. In such cases, the ID copy can be accessed by unauthorized parties, who can use the information for nefarious purposes, such as identity theft or financial fraud.

Furthermore, even if the app itself is not compromised, it may still collect and transmit personal data to third-party servers without the user's knowledge or consent, posing a significant threat to privacy. In Jan 2021, a person was sentenced to five years in prison after using a state-authorized digital driver's license mobile app to defraud three credit unions and four banks [11]. According to a recent report [12] from McAfee, 15 million Americans had their identity stolen in 2021. Based on data gathered by Finanso.se [13], one in five Europeans have experienced identity theft fraud between 2020 and 2022. In 39% of these cases, the attackers used the victim's phone to steal their identity [13].

To address these key concerns, app providers must follow security and privacy best practices during the app's lifecycle, aiming to prevent theft of sensitive personal information. Regarding the Android ecosystem, there exist several noteworthy standards and best practices for developing secure software, including the Android developer website [14], the Open Worldwide Application Security Project (OWASP) mobile top 10 project [15], and the Japan Smartphone Security Association (JSSEC) Android application secure design/secure coding guidebook [16]. Additionally, there are several works in the literature that focus on the security of the Android OS, such as [17,18]. However, compliance to these practices is often limited by the complexity of the underlying technology, the developer's level of technical expertise, and the lack of standardized security protocols and policies. Moreover, the development of such apps may be outsourced to third parties who perform these functions on behalf of the solution provider. In such a case, trust is indirect, and sometimes cannot be fully assessed.

The present work focuses on mobile wallets that support IDs and DLs, offering the first, to our knowledge, exhaustive review and examination of this topic based on three app security analysis tools. Precisely, the contributions of this study are as follows:

- We present an overview of the existing official mobile apps supporting IDs and DLs, as well as the privacy and security risks associated with storing digital ID and DL documents. The term "official" means apps that are either offered by governmental agencies (state-sponsored) or by a mobile operating system (OS), say, Android or iOS.
- We collect and analyze existing Android apps for ID and DL storage using three *.apk* analysis tools, present the discovered vulnerabilities of each app, and discuss key findings.
- We offer recommendations for app developers and relevant stakeholders to enhance the privacy and security of ID and DL storage in Android apps.

The rest of this paper is structured as follows. The next section surveys official ID/DL wallet apps. Section 4 presents the vulnerability analysis results. Section 5 details the key findings of the previous section and provides recommendations to improve the security status of the analyzed apps. The last section concludes the paper.

2. Related Work

The domain of ID/DL wallet apps is rather new and, to the best of our knowledge, there is no previous work tackling the issue of ID/DL wallet app security and privacy. There is, however, a large volume of work of a similar nature evaluating generic Android app security and privacy.

Filiol and Irola [19] analyzed numerous mobile apps in the banking domain. The authors showed that almost all apps were prone to known vulnerabilities, endangering users' private data, sometimes severely. The authors also discussed the certification process for apps available on a secure market. Kaur et al. [20] presented a security assessment of the Android e-wallet apps provided by Canada's leading banks. According to their analysis, all apps were found to be vulnerable against trivial attack vectors.

In the health domain, Papageorgiou et al. [7] provided a security and privacy analysis of popular freeware mobile health apps. The authors employed both static and dynamic analysis, as well as custom testing of each application. Their analysis demonstrated that the majority of apps neither follow well-known practices and guidelines nor comply with data protection regulations. Kouliaridis et al. [8] focused on contact tracing apps used for decelerating the spread of infectious diseases. They analyzed all official Android contact tracing apps deployed by European countries by means of dynamic instrumentation. Their findings revealed that these apps may put users' security and privacy at risk due to an assortment of weaknesses, vulnerabilities, and misconfigurations. Karopoulos et al. [9] examined existing initiatives for COVID-19 digital certificates undertaken by organizations and countries worldwide. As part of their study, they analyzed official Android apps for COVID-19 digital certificates to reveal possible security and privacy issues affecting the end user. Their results demonstrated that, overall, the schemes developed by European countries provide a higher level of privacy protection compared to those from Asia and America.

In the automotive domain, Mandal et al. [21] analyzed Android infotainment apps against a list of possible exposure scenarios. Their results showed that almost 80% of these apps were potentially vulnerable. Chatzoglou et al. [10] provided a security assessment of all the official car management apps offered by major car manufacturers operating in Europe. The apps were assessed for vulnerabilities and possible weak security practices. Their analysis reported numerous issues, ranging from privacy-invasive permissions and API calls, to potentially exploitable common weakness enumeration (CWE) and common vulnerabilities and exposures (CVE)—identified weaknesses and vulnerabilities.

Regarding the use of cryptography, Egele et al. [22] developed an automatic analysis technique to find Android apps on Google Play that use cryptographic APIs. The authors reported that 88% of these apps misused cryptographic APIs, making at least one mistake that resulted in decreasing the maximum achievable security level. To this end, they provided recommendations to improve the cryptographic security of such apps. Chatzikonstantinou et al. [23] evaluated the use of cryptography in 49 Android apps whose operation is related to data encryption. Their results revealed that the majority of these apps, i.e., around 88%, presented at least one type of cryptographic weakness. The authors provided guidelines and best practices for developers, to aid in the development of more secure apps.

More recently, Chatzoglou et al. [24] performed a fully fledged analysis of more than 40 mainstream internet of things (IoT) official Android apps belonging to six popular categories of home/office and wearable devices. They pinpointed that most of the examined apps were susceptible to an assortment of security and privacy issues, including transmission of cleartext traffic, outdated software components, no protection against reverse engineering, and others. They concluded that the attack surface for an IoT device is significantly augmented because of the security weaknesses in the accompanying app.

Although more and more manufactures are relying on trusted execution environments (TEEs) to shield their devices, Ref. [25] provides an extensive analysis and categorization of existing vulnerabilities in TEEs and shows the design flaws that lead to them. The authors in [26], released new state of the art mobile app datasets along with an in-depth analysis of their static characteristics to aid the detection of Android malware with the use of both shallow and deep learning techniques.

The objective of the above summary of analyses of Android apps is to highlight the main issues related to the security and privacy of different types of apps. Overall, previous work in the field underlines that even officially certified Android apps, also under the scrutiny of the official Google Play app store, present numerous issues that can potentially endanger users' security and privacy. In the rest of this paper, we perform similar analyses to investigate whether this holds true for recently launched ID/DL apps as well, given that this is still an unexplored field.

3. ID/DL Wallet Apps Worldwide

As already pointed out in Section 1, ID/DL wallet apps can be classified in two main categories: either state-sponsored or offered by a mobile operating system (OS). The former category of wallet apps are developed under the auspices of the government of a specific country or state. Apart from state-sponsored apps, the main mobile OS platforms, that is, Android and iOS that together account for more than 99% of the respective market share [27,28], have announced support for mobile IDs and/or mobile DLs. The fact that both of these platforms are active in the domain of mobile ID/DL is a key factor towards the wide adoption of such solutions.

Looking at the current support by mobile platforms, in December 2022, Google announced support for storing state IDs and DLs from selected US states in Google Wallet as a beta feature [29]. On the iOS side, Apple announced in 2021 that some US states had signed up to make available state IDs and DLs in Apple Wallet [30]. According to the US Transportation Security Administration (TSA) [3], various airports around the US currently accept mobile IDs and DLs stored in Apple Wallet issued by the Arizona, Colorado, and Maryland states.

It should be noted here that mobile-OS-supported IDs and DLs could possibly entail similar security risks as ID wallet apps. More specifically, in some use cases, the service might require unlocking the smartphone to access the ID/DL and handing out the unlocked device to the interested, authorized party, i.e., police or other public or private service agent. On the other hand, if the electronic ID and DL are available without the need to unlock the phone, the personal information contained in them will be visible to anyone who picks up the device. A balanced use case scenario between security and usability would provide access to the electronic ID/DL using biometric authentication, without unlocking the smartphone.

In this work, we only consider official, state-sponsored ID/DL wallet apps for the Android OS. To our knowledge, the 18 official ID/DL wallet apps available as of the time of writing of this paper are those listed in Table 1.

Table 1. Outline of the examined apps (ID: identity document, DL: driver's license).

Country/State	App Name	ID	DL	Downloads	Android Version	App Providers
North America						
Louisiana, USA	LA wallet [31]	Yes	Yes	500 K	5.0+	State of Louisiana
Colorado, USA	myColorado [32]	Yes	No	100 K	8.1+	State of Colorado—Governor's Office of IT
Florida, USA	FL Smart ID: Thales [33]	No	Yes	10 K	6.0+	Florida Department of Highway Safety and Motor Vehicles
Georgia, USA	DDS 2 GO [34]	No	Yes	500 K	5.1+	Georgia Department of Driver Services
Oklahoma, USA	Oklahoma Mobile ID [35]	Yes	No	100 K	6.0+	Idemia R&D
Delaware, USA	Delaware Mobile ID [36]	Yes	No	10 K	6.0+	Idemia R&D
Utah, USA	GET Mobile ID [37]	Yes	Yes	10 K	8.0+	GET Group NA
USA	Airside Digital Identity [38]	Yes	Yes	10 K	8.0+	American Airlines/Airside Mobile Inc.
Canada	eID-Me Digital ID [39]	Yes	No	10 K	8.0+	Bluink Ltd.

Table 1. *Cont.*

Country/State	App Name	ID	DL	Downloads	Android Version	App Providers
Europe						
Austria	eAusweise [40]	No	Yes	100 K	8.0+	Bundesministerium für Finanzen
Denmark	Kørekort [41]	No	Yes	500 K	8.0+	Digitaliseringsstyrelsen
Germany	Verimi ID wallet [42]	Yes	Yes	100 K	7.0+	Verimi
Greece	Gov.gr Wallet [43]	Yes	Yes	500 K	8.0+	Hellenic Republic
Netherlands	KopieID [44]	Yes	No	1 M	7.0+	Rijksoverheid
Portugal	id.gov.pt [45]	Yes	Yes	500 K	4.2+	AMA, IP
Spain	mi DGT [46]	No	Yes	5 M	5.1+	DGT oficial
Asia						
Telangana, India	RTA m-wallet [47]	No	Yes	5 K	5.0+	Transport Department Govt. of Telangana
Oceania						
Australia	Service NSW [48]	No	Yes	1 M	6.0+	Service NSW

4. Vulnerability Analysis

The aim of this section is to present key results regarding the vulnerability analysis of the wallet apps given in Table 1. Specifically, the 18 ID/DL wallet apps were collected from Google Play with a freeze date of 1 June 2023. Each of them was statically analyzed using three tools, namely, Ostorlab [49], Mobile Security Framework (MobSF) [50], and Androtomist [51]. The detailed results of the security assessment performed with the aforementioned tools can be found in [52].

Ostorlab is a cloud-based security platform that caters for dynamic and static analysis of mobile apps. It allows users to scan an app for vulnerabilities, such as insecure injection, outdated dependencies, hardcoded secrets, weak cryptography, cleartext communication, configuration issues, and improper use of permissions. The tool also provides a detailed report of the findings, including the severity of each vulnerability, i.e., *low*, *medium*, or *high*. Moreover, it provides recommendations for remediation. According to the tool's web page, more than 10K companies and security professionals rely on it for Android app penetration testing. The overall risk rating of the app is calculated by aggregating the individual ratings of each vulnerability. More specifically, Ostorlab uses the following techniques to find vulnerabilities:

- Configuration checks for insecure settings. These settings include Android native parameters, e.g., in the AndroidManifest.xml.
- Third-party dependency analysis to find all application dependencies of all supported frameworks, as well as statically compiled dependencies, and identify a large set of libraries. The tool then tries to match these libraries against its known vulnerabilities database.
- Hardcoded secrets scanning, i.e., API keys, passwords, tokens, encryption keys, and initialization vectors (IVs).
- Taint analysis to identify vulnerabilities, such as SQL injection, command injection, or the use of hardcoded keys.

In contrast to MobSF, Ostorlab reports the use of outdated dependencies. Additionally, Ostorlab also checks supply chain vulnerabilities, such as dependency confusion, namely, attacks directed against third-party dependencies in an app. Recall that third-party dependencies refer to libraries, frameworks, and other software built by external parties and are embedded into the app.

MobSF is one of the all-in-one tools recommended by the OWASP Mobile Security Testing Guide [53]. MobSF is a popular open-source mobile app security testing framework that allows users to perform static and dynamic analysis of Android apps. The static analysis includes source code, binary, tracker analysis, and configuration analysis, while the dynamic analysis is based on runtime behavior analysis, code injection, and traffic interception. The tool can be used to identify vulnerabilities, such as sensitive data disclosure, insecure cryptography, and insecure communications. It also provides detailed reports on the findings, including a list of vulnerabilities, their respective CWE, and a score using the common vulnerability scoring system (CVSS), i.e., 0–3.9 = low, 4–6.9 = medium, and 7–10 = high. To compute an overall score for the app, first, a severity level, high, warning, or good, is assigned to each vulnerability by MobSF. The final score of the app is calculated by first assigning a perfect score of 100 and then for each vulnerability applying the following:

- severity high—subtracting 15 from the score;
- severity warning—subtracting 10 from the score;
- severity good—adding 5 to the score.

Apart from the above-mentioned well-known tools, the authors used an self-developed tool that, however, has already been used in relevant research. The reason for using this tool in conjunction with the other two is that it gave us more fine-grained control over the analysis process. Androtomist is an automated and configurable tool, which combines static and dynamic analysis to evaluate Android app behavior. In the context of this paper, it has been used to statically analyze each app and extract components from the manifest file, such as activities, services, and broadcast receivers. Activities are used when one app invokes a component of another app instead of calling the whole app. For example, a social media app can call the email composer component of an email app. However, an activity constitutes a potential entry point for malicious entities if not properly secured, increasing the attack surface of the app. A service, on the other hand, is an app component that runs in the background without providing a user interface, such as a service handling network tasks, playing music, or performing file I/O operations. Furthermore, a service can remain active even when the user switches to another application. Broadcast receivers are used to send and receive messages between apps, such as notifications or alarms. If an app's broadcast receiver is not secured properly, it may allow other apps to intercept and read the messages. This can lead to sensitive information being leaked, such as passwords or personal data. Finally, Androtomist employs static taint analysis, which aids in finding complex vulnerabilities spanning long code paths.

By using three separate tools, this work aims to provide a comprehensive understanding of the security and privacy level of the examined Android apps. Our results rely on static analysis only and focus on code vulnerabilities; Table 2 summarizes the analysis results per tool. Specifically, the table presents Ostorlab's risk raking, MobSF's security score, and the number of exported activities, services, and receivers reported by the Androtomist tool.

Table 2. Vulnerability analysis results: high risk and low risk security scores have been emphasized with bold font.

App Name	Ostorlab Risk Rating	MobSF Security Score (%)	Exported Activities-Services-Receivers
LA wallet	**High**	Medium (45)	1-4-1
myColorado	**High**	Medium (53)	1-0-1
FL Smart ID: Thales	**High**	Medium (57)	1-1-1
DDS 2 GO	Medium	**Low (38)**	2-1-1
Oklahoma Mobile ID	**High**	Medium (60)	3-0-3

Table 2. *Cont.*

App Name	Ostorlab Risk Rating	MobSF Security Score (%)	Exported Activities-Services-Receivers
Delaware Mobile ID	**High**	Medium (55)	3-0-2
GET Mobile ID	**High**	High (69)	3-3-1
Airside Digital Identity	Low	Medium (62)	2-2-2
eID-Me Digital ID	**High**	Medium (56)	1-1-1
eAusweise	Low	Medium (60)	6-0-1
Kørekort	Low	Medium (60)	0-1-1
Verimi ID wallet	Medium	Medium (64)	1-2-2
Gov.gr Wallet	**High**	Medium (56)	2-2-2
KopieID	Low	Medium (62)	1-0-0
id.gov.pt	**High**	Medium (51)	1-0-1
mi DGT	**High**	Medium (51)	7-1-2
RTA m-wallet	**High**	Medium (44)	0-1-1
Service NSW	**High**	Medium (44)	2-2-5

5. Discussion

This section wraps up our key findings from the vulnerability analysis of Section 4, for each of the three tools, namely, Ostorlab, MobSF, and Androtomist. Recall that the analytical results per tool can be found in [52].

5.1. Ostorlab

In order to provide a global overview, we summarize the top vulnerabilities identified by Ostorlab and MobSF in Figure 1; in the rest of this subsection, we analyze our findings with Ostorlab. Our analysis showed 47 cases of use of outdated vulnerable components in 14 apps, which can be exploited by malicious parties; most of these cases have a high risk rating. Third-party libraries should be updated to the latest version during the development phase and application updates should be issued to patch vulnerable components. Regarding cryptography use, all but one of the apps were found not to follow best practices by using hardcoded keys, storing secret information in the app, using non-random or insecure random values, supporting deprecated cipher suites, or performing incorrect certificate validation. Note that insufficient cryptography is placed in the fifth position of the latest OWASP top 10 mobile risks list [15]. In addition, three apps, i.e., "DDS 2 GO", "Service NSW", and "mi DGT" have set the *usesCleartextTraffic* attribute to "true", which indicates that the apps intend to exchange or allow cleartext network traffic. In the OWASP top 10 mobile risks list, insecure communication is placed in the third position. Obviously, given that such apps are used for storing ID/DL documents, cleartext network traffic could allow data theft over the network simply by means of packet sniffing. In total, 12, or approximately 67%, of the apps have a high risk rating according to Ostorlab.

Notably, all apps have been flagged with the "task highjacking" warning [54]. Task hijacking can be used to perform phishing attacks. This is a noteworthy issue, as an attacker could potentially capture and read triggered intents. For example, CVE-2020-0096, also known as "Standhogg 2.0", can potentially exploit this issue in unpatched Android OS v8, 8.1, and 9. According to Android's guidelines for "task affinities" [55], setting the "android:launchMode" attribute in the <activity> tag to "singleInstance" forbids other activities to be part of its task. Furthermore, setting the "android:taskAffinity" attribute to an empty string in the <activity> tag forces the activities to use a randomly generated task affinity. Last but not least, by using explicit intents, developers can specify which application will satisfy the intent. In addition, approximately 61% of the apps were flagged with the "intent spoofing" warning [56]. This vulnerability can be exploited by sending an intent towards an app's exported component, i.e., activity, receiver, or service, to obtain unauthorized access. Each exported component should check the caller's identity prior to executing any tasks. Ostrolab also suggests requiring *signature* or *signatureorsystem* level

permissions to limit a component's exposure to a set of trusted applications [56]. Finally, there were many warnings flagging potential risks; in most cases, these warnings are false positives or do not pose a significant risk. Nevertheless, developers should examine these cases as well to identify potential security or privacy issues.

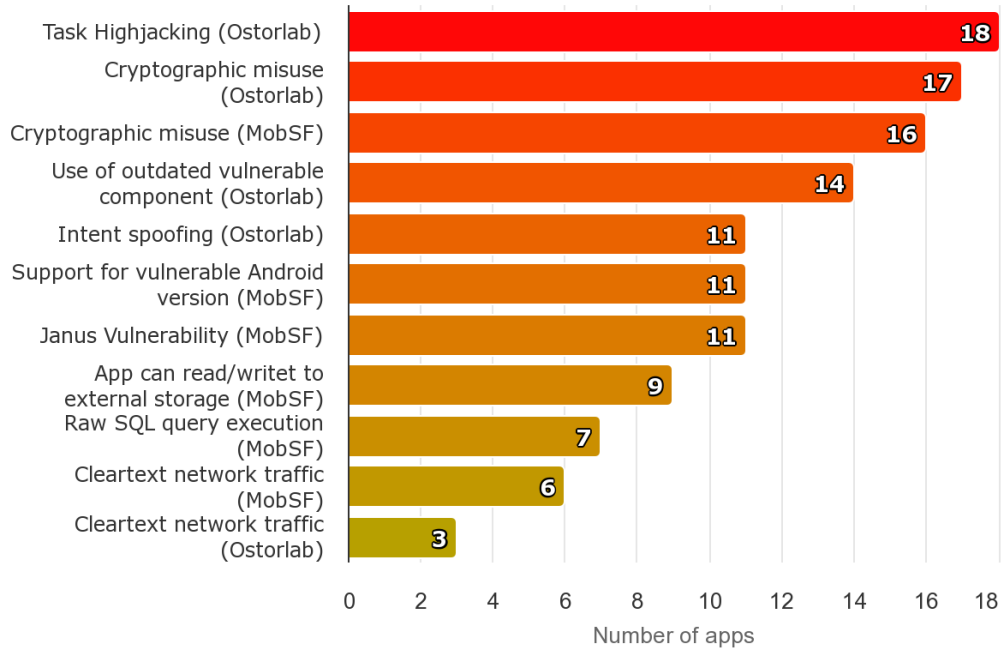

Figure 1. Top vulnerabilities as reported by Ostorlab and MobSF.

5.2. MobSF

As already mentioned in Section 4, in contrast to Ostorlab, MobSF provides a security score, where a higher score indicates a more secure app. Overall, out of the 18 apps, only "GET Mobile ID" received a low security risk score (>71%), 16 apps were granted a medium score (41%–70%), and "DDS 2 GO" received a high risk score (<40%); an overview of the results is presented in Figure 2.

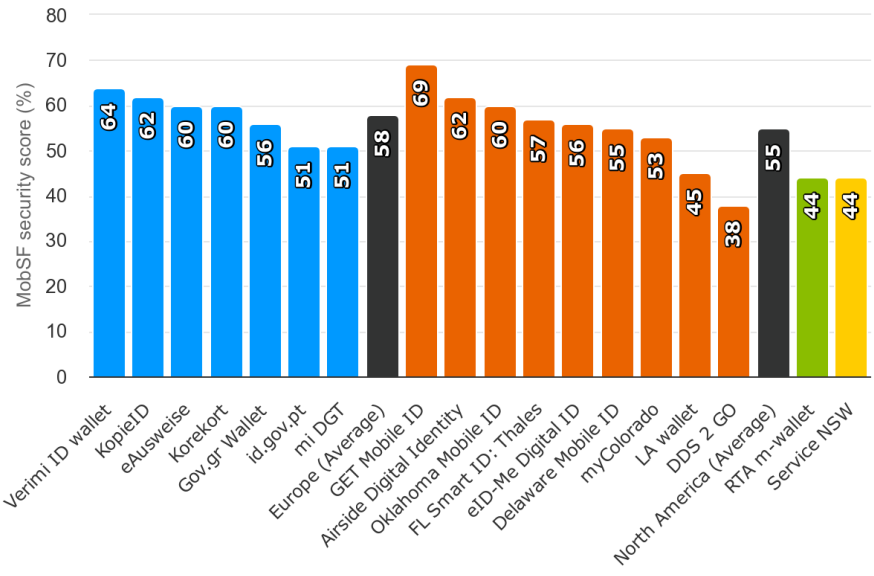

Figure 2. MobFS security score results, a higher score indicates a more secure app (blue: Europe, orange: North America, green: Asia, yellow: Oceania).

As noted earlier, a summary of the top vulnerabilities identified by both Ostorlab and MobSF is provided in Figure 1; in the rest of this subsection we analyze our findings with MobSF. According to MobSF, more than half (11) of the apps appear to be vulnerable to the so called "Janus vulnerability", documented in CVE-2017-13156 [57]. This vulnerability allows attackers to modify apps without affecting their signatures, i.e., adding extra bytes to the android package kit (APK) and DEX (Dalvik virtual machine executable) files. However, it only affects Android devices before v8.1, when signed with the v1 signature scheme. A similarly high proportion of apps, approximately 56% (10 out of 18), can be installed on a vulnerable Android version. Furthermore, all but two apps present at least one cryptographic misuse or warning; the exceptions are "mi DGT" and "Oklahoma Mobile ID". In addition, one-third of the apps use SQLite and execute raw SQL queries, which could lead to SQL injection attacks. Another important finding was that six of the apps allowed cleartext traffic in general or to/from specific network domains or IP addresses. As already stated above, in the OWASP top 10 mobile risks list, this warning is placed in the third position. Finally, 14 out of the 18 apps received the "insertion of sensitive information into log file" warning (CWE-532 [58]). While logging information is helpful during the development stage of an app, it must be stripped away before the app becomes publicly available. Precisely, an attacker could analyze the logs to extract private information stored on them. Finally, 50% of the apps received the "insecure data storage" warning (CWE-276) as they can read/write to external storage. This can be dangerous as any app can read data written to external storage. This warning is placed in the second position in the OWASP top 10 mobile risks list.

MobSF also logs third-party trackers that may be utilized by each app. We focus on six common tracker categories.

- Crash reporters: These trackers notify developers upon a crash event, informing them about the respective error.
- Analytics trackers: Collect usage information, e.g., time users spent on the app and top features used.
- Profiling trackers: Attempt to profile users with the purpose of optimizing personalized advertising.
- Identification trackers: Gather information with the purpose of ultimately matching a digital (user) identity with the real person.
- Ads: These trackers focus on serving personalized advertisements to the users.
- Location trackers: By using location services, these trackers obtain the geographical location of the user to improve location-based personalized advertisements.

As shown in Figure 3, app analysis revealed that all but three apps use trackers. The exceptions are "eAusweise", "Verimi ID wallet", and "GET Mobile ID". On the other hand, 13 apps use the *Firebase* Google analytics service as a method to measure users' engagement with them. Furthermore, seven apps exploit Google crashLytics to track code issues and app crashes. Additionally, "Delaware Mobile ID" and "Oklahoma Mobile ID" use AppsFlyer, which tracks all app-related events that are generated by clients, to improve personalized advertisements, as well as Localytics, which is a marketing tool used to engage users via targeted push and in-app messages (ads). Last but not least, "KopieID" uses Countly, which tracks and analyzes user behavior.

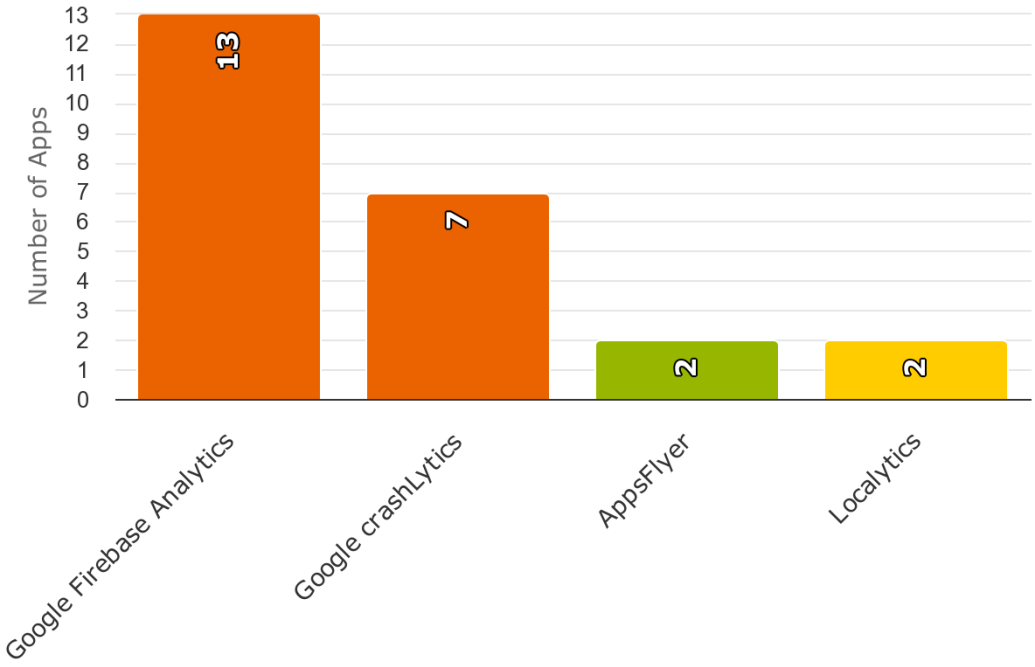

Figure 3. Top trackers used.

5.3. Androtomist

Finally, Androtomist was used to decompile each app, extract their manifest file, and log their exported components. In addition, taint analysis was also performed to extract possible data leaks. According to our results, none of the analyzed apps has exported content providers. On the other hand, two-thirds of the apps have at least one exported service (exceptions are "myColorado", "Oklahoma Mobile ID", "Delaware Mobile ID", "eAusweise", "KopieID", and "ip.gov.pt"). Furthermore, 16 apps have at least one exported activity (exceptions are "RTA m-wallet" and "Kørekort") and all apps except "KopieID" have at least one exported receiver.

When comparing the exported components of each, it is noted that "mi DGT" and "eAusweise" have seven and six exported activities, respectively, while the rest of the apps have three or less exported activities. Moreover, "LA wallet" has the most (four) exported services, while "Service NSW" has the most (five) exported receivers.

To prevent data leaks through broadcast receivers, app developers should implement appropriate security measures, such as setting proper permissions, restricting access to sensitive data, and using encrypted communication channels. Android end users should also be cautious when granting permissions to apps and limit access to sensitive data whenever possible. Android apps can set exported components, i.e., components that can be used by other applications, but often do not properly restrict which applications can launch the component or access the data they contain [59]. Additionally, we employed taint analysis on all apps and our results did not reveal any leaks.

5.4. Key Takeaways

As shown in Table 2, Europe has a lower percentage of high risk apps and a higher average security score than North America, as measured with Ostorlab. Specifically, three out of seven apps have a high risk rating in Europe, compared to seven out of nine apps in North America. Similarly, based on MobSF's results, Europe performs better with a security score of 58% compared to 55% of North American apps, as shown in Table 2 and Figure 2. Equally important is the number of CWEs reported by MobSF, i.e., 4 CWEs per application on average in Europe vs. 5.2 CWEs per app on average in North America.

With reference to Figure 1, which depicts the top vulnerabilities from both MobFS and Ostorlab, it is apparent that the tools are complementary to each other, reporting some common as well as unique findings. Furthermore, MobFS scores the security of apps, while Ostorlab measures the opposite, i.e., their security risk. As such, the use of both tools played an important role in identifying a variety of warnings and vulnerabilities.

It is also important here to comment on the overall results presented in Table 2. As it has already been briefly discussed previously, the three analysis tools assess different aspects of each app. Ostorlab has three extra analysis sections, including taint analysis, which makes it report more information. Moreover, Ostorlab reports risk ratings, meaning that even if an app has a single high risk vulnerability, then it is considered a high risk app. MobSF reports security scores by adding or subtracting points from a base score of 100%. This means that, on the one hand, identified vulnerabilities decrease but, on the other, good practices increase the score. For this reason, it is quite rare for an app to receive a low security score, unless it is a malware. Regarding Androtomist, the exported components used in the context of this work are only a small part of the analysis, whereas the other tools perform a much deeper analysis. Summing up, to achieve an overall idea of the security posture of an app, one should consider the results of all the three tools in a combinatorial manner in order to obtain an approximation of the total level of risk.

6. Conclusions

This paper conducted a comprehensive analysis concentrating on security and privacy aspects of the so far available ID/DL wallet apps. In other words, we attempt to answer the key question: Are these apps free of vulnerabilities which are known to the community, say, already documented in a CVE ID? To this end, three different software tools were used to analyze such apps and identify vulnerabilities. Our findings revealed significant (even critical but straightforward) security flaws that considerably increase the attack surface and could severely undermine the overall end user's security and privacy. Additionally, suggestions for app developers that enhance the security of these apps were discussed. It can be said that the overall picture is not so encouraging, suggesting that app creators and other stakeholders should devote more attention to security and privacy, not treating them as an afterthought. Actually, this tendency in tossing security aside, typically in favor of functionality, is corroborated by the related work as detailed in Section 2. Notably, the Android platform is currently working on its own ID wallet service [29], which could serve as an alternative solution for governments considering ID wallet app development. Future research should include a security evaluation of this component. Furthermore, cybersecurity policies such as those introduced by the European Commission [60] can provide guidance to member states in developing more secure and resilient solutions.

Author Contributions: Conceptualization, V.K. and G.K. (Georgios Karopoulos); data curation, V.K.; formal analysis, V.K., G.K. (Georgios Karopoulos), and G.K. (Georgios Kambourakis); investigation, V.K. and G.K. (Georgios Karopoulos); methodology, V.K., G.K. (Georgios Karopoulos), and G.K. (Georgios Kambourakis); resources, V.K. and G.K. (Georgios Kambourakis); software, V.K.; supervision, G.K. (Georgios Kambourakis); validation, G.K. (Georgios Karopoulos) and G.K. (Georgios Kambourakis); visualization, V.K., G.K. (Georgios Karopoulos) and G.K. (Georgios Kambourakis); writing—original draft, V.K. and G.K. (Georgios Karopoulos); writing—review and editing, V.K., G.K. (Georgios Karopoulos) and G.K. (Georgios Kambourakis). All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Results from our analysis can be found at https://github.com/billkoul/AndroidIDWalletApps (accessed on 4 July 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

APK	Android application package
CVE	Common vulnerabilities and exposures
CWE	Common weakness enumeration
DL	Driver's license
ID	Identity document
IoT	Internet of things
IV	Initialization vectors
JSSEC	Japan smartphone security association
OWASP	Open worldwide application security project
SQL	Structured query language

References

1. European Commission. European Digital Identity. Available online: https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-digital-identity_en (accessed on 4 July 2023).
2. European Commission. Road Safety: Commission Proposes Updated Requirements for Driving Licences and Better Cross-Border Enforcement of Road Traffic Rules. Available online: https://ec.europa.eu/commission/presscorner/detail/en/ip_23_1145 (accessed on 4 July 2023).
3. Transportation Security Administration . When Will the Phased Digital ID Rollout Start? Which Airports/States Will Be First in Line for This New Technology? Available online: https://www.tsa.gov/travel/frequently-asked-questions/when-will-phased-digital-id-rollout-start-which-airportsstates (accessed on 4 July 2023).
4. GLOBE NEWSWIRE. Study: More than Half of the World's Population Will Use Mobile Wallets by 2025. Available online: https://www.globenewswire.com/en/news-release/2021/07/08/2259605/0/en/Study-More-than-half-of-the-world-s-population-will-use-mobile-wallets-by-2025.html (accessed on 4 July 2023).
5. Damopoulos, D.; Kambourakis, G.; Anagnostopoulos, M.; Gritzalis, S.; Park, J.H. User privacy and modern mobile services: are they on the same path? *Pers. Ubiquitous Comput.* **2013**, *17*, 1437–1448. [CrossRef]
6. Papamartzivanos, D.; Damopoulos, D.; Kambourakis, G. A cloud-based architecture to crowdsource mobile app privacy leaks. In Proceedings of the 18th Panhellenic Conference on Informatics, PCI '14, Athens, Greece, 2–4 October 2014; ACM: New York, NY, USA, 2014, pp. 59:1–59:6. [CrossRef]
7. Papageorgiou, A.; Strigkos, M.; Politou, E.; Alepis, E.; Solanas, A.; Patsakis, C. Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice. *IEEE Access* **2018**, *6*, 9390–9403. [CrossRef]
8. Kouliaridis, V.; Kambourakis, G.; Chatzoglou, E.; Geneiatakis, D.; Wang, H. Dissecting contact tracing apps in the Android platform. *PLoS ONE* **2021**, *16*, 1–28. [CrossRef]
9. Karopoulos, G.; Hernandez-Ramos, J.L.; Kouliaridis, V.; Kambourakis, G. A Survey on Digital Certificates Approaches for the COVID-19 Pandemic. *IEEE Access* **2021**, *9*, 138003–138025. [CrossRef]
10. Chatzoglou, E.; Kambourakis, G.; Kouliaridis, V. A Multi-Tier Security Analysis of Official Car Management Apps for Android. *Future Internet* **2021**, *13*, 58. [CrossRef]
11. Louisiana Man Uses Digital Driver's License to Defraud Credit Unions & Banks. Available online: https://www.cutimes.com/2023/03/16/louisiana-man-uses-digital-drivers-license-to-defraud-credit-unions-banks/?slreturn=20230708061731 (accessed on 4 July 2023).
12. A Guide to Identity Theft Statistics for 2023. Available online: https://www.mcafee.com/learn/a-guide-to-identity-theft-statistics/ (accessed on 4 July 2023).
13. One in Five Europeans Have Experienced Identity Theft Fraud in the Last Two Years. Available online: https://finanso.se/one-in-five-europeans-have-experienced-identity-theft-fraud-in-the-last-two-years/ (accessed on 4 July 2023).
14. Android. App Security Best Practices. Available online: https://developer.android.com/topic/security/best-practices (accessed on 4 July 2023).
15. OWASP Mobile Top 10. Available online: https://owasp.org/www-project-mobile-top-10/ (accessed on 4 July 2023).
16. jssec. Android Application Secure Design/Secure Coding Guidebook. Available online: https://www.jssec.org/dl/android_securecoding_en_20220117/index.html (accessed on 2022 4 July 2023).

17. Garg, S.; Baliyan, N. Comparative Analysis of Android and IOS from Security Viewpoint. *Comput. Sci. Rev.* **2021**, *40*, 100372. [CrossRef]
18. Sarkar, A.; Goyal, A.; Hicks, D.; Sarkar, D.; Hazra, S. Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems. In Proceedings of the 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 73–79. [CrossRef]
19. Filiol, E.; Irolla, P. Security of Mobile Banking… and of Other Mobile Apps. In Proceedings of the Black Hat Asia, Singapore, 24–27 March 2015; pp. 1–22.
20. Kaur, R.; Li, Y.; Iqbal, J.; Gonzalez, H.; Stakhanova, N. A Security Assessment of HCE-NFC Enabled E-Wallet Banking Android Apps. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 02, pp. 492–497. [CrossRef]
21. Mandal, A.K.; Cortesi, A.; Ferrara, P.; Panarotto, F.; Spoto, F. Vulnerability analysis of android auto infotainment apps. In Proceedings of the 15th ACM International Conference on Computing Frontiers, Ischia, Italy, 8–10 May 2018; pp. 183–190.
22. Egele, M.; Brumley, D.; Fratantonio, Y.; Kruegel, C. An Empirical Study of Cryptographic Misuse in Android Applications. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; Association for Computing Machinery: New York, NY, USA, 2013; p. 73–84. [CrossRef]
23. Chatzikonstantinou, A.; Ntantogian, C.; Karopoulos, G.; Xenakis, C. Evaluation of Cryptography Usage in Android Applications. *EAI Endorsed Trans. Secur. Saf.* **2016**, *3*, e4. [CrossRef]
24. Chatzoglou, E.; Kambourakis, G.; Smiliotopoulos, C. Let the Cat out of the Bag: Popular Android IoT Apps under Security Scrutiny. *Sensors* **2022**, *22*, 513. [CrossRef] [PubMed]
25. Muñoz, A.; Ríos, R.; Román, R.; López, J. A survey on the (in)security of trusted execution environments. *Comput. Secur.* **2023**, *129*, 103180. [CrossRef]
26. Gómez, A.; Muñoz, A. Deep Learning-Based Attack Detection and Classification in Android Devices. *Electronics* **2023**, *12*, 3253. [CrossRef]
27. Statista. Mobile Operating Systems' Market Share Worldwide from 1st Quarter 2009 to 4th Quarter 2022. Available online: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/ (accessed on 4 July 2023).
28. Statcounter. Mobile Operating System Market Share Worldwide. Available online: https://gs.statcounter.com/os-market-share/mobile/worldwide (accessed on 4 July 2023).
29. Google Inc.. What's New in Google System Updates. Available online: https://support.google.com/product-documentation/answer/11412553 (accessed on 2022 4 July 2023).
30. Apple Inc.. Apple Announces First States Signed Up to Adopt Driver's Licenses and State IDs in Apple Wallet. Available online: https://www.apple.com/newsroom/2021/09/apple-announces-first-states-to-adopt-drivers-licenses-and-state-ids-in-wallet/ (accessed on 4 July 2023).
31. Lawallet App. Available online: https://play.google.com/store/apps/details?id=gov.la.omv.lawallet (accessed on 4 July 2023).
32. MyColorado App. Available online: https://play.google.com/store/apps/details?id=com.soc.mycolorado (accessed on 4 July 2023).
33. FL Smart ID App. Available online: https://play.google.com/store/apps/details?id=com.thalesgroup.dis.idv.fl.holder.prd (accessed on 4 July 2023).
34. dds2go App. Available online: https://play.google.com/store/apps/details?id=gov.ga.dds.gadds (accessed on 4 July 2023).
35. Oklahoma Mobile ID App. Available online: https://play.google.com/store/apps/details?id=com.idemia.mobileid.us.ok (accessed on 4 July 2023).
36. Delaware Mobile ID App. Available online: https://play.google.com/store/apps/details?id=com.idemia.mobileid.us.de (accessed on 4 July 2023).
37. GET Mobile ID. Available online: https://play.google.com/store/apps/details?id=com.getgroupna.mdl.app.utah (accessed on 4 July 2023).
38. Airside Digital Identity. Available online: https://play.google.com/store/apps/details?id=com.airsidemobile.digitalid.android.prod (accessed on 4 July 2023).
39. eID-Me Digital ID App. Available online: https://play.google.com/store/apps/details?id=ca.bluink.eid_me_and (accessed on 4 July 2023).
40. eAusweise App. Available online: https://play.google.com/store/apps/details?id=at.gv.oe.awp.eausweise (accessed on 4 July 2023).
41. Kørekort app. Available online: https://play.google.com/store/apps/details?id=dk.digst.mdl (accessed on 4 July 2023).
42. Verimi ID Wallet App. Available online: https://play.google.com/store/apps/details?id=com.verimi (accessed on 4 July 2023).
43. gov.gr App. Available online: https://play.google.com/store/apps/details?id=gr.gov.wallet (accessed on 4 July 2023).
44. Kopie ID App. Available online: https://play.google.com/store/apps/details?id=com.milvum.kopieid (accessed on 4 July 2023).
45. id.gov.pt App. Available online: https://play.google.com/store/apps/details?id=id.gov.pt (accessed on 4 July 2023).
46. mi DGT App. Available online: https://play.google.com/store/apps/details?id=com.dgt.midgt&hl=en (accessed on 4 July 2023).

47. RTA m-Wallet App. Available online: https://play.google.com/store/apps/details?id=tsgovt.com.mywalet (accessed on 4 July 2023).
48. Service NSW App. Available online: https://play.google.com/store/apps/details?id=au.gov.nsw.service (accessed on 4 July 2023).
49. Ostorlab. Mobile Application Security Testing. Available online: https://www.ostorlab.co/product/mobile (accessed on 4 July 2023).
50. Abraham, A.; Schlecht, D.; Dobrushin, M.; Nadal, V. Mobile security framework (MobSF). Available online: https://github.com/MobSF/Mobile-Security-Framework-MobSF (accessed on 4 July 2023).
51. Kouliaridis, V.; Kambourakis, G.; Geneiatakis, D.; Potha, N. Two Anatomists Are Better than One-Dual-Level Android Malware Detection. *Symmetry* **2020**, *12*, 1128. [CrossRef]
52. Android Official ID Wallet Apps–Analysis Results. Available online: https://github.com/billkoul/AndroidIDWalletApps (accessed on 4 July 2023).
53. OWASP Mobile App Security. Available online: https://owasp.org/www-project-mobile-app-security/ (accessed on 4 July 2023).
54. Task Hijacking. Available online: https://docs.ostorlab.co/kb/APK_TASK_HIJACKING/ (accessed on 4 July 2023).
55. Handle Affinities. Available online: https://developer.android.com/guide/components/activities/tasks-and-back-stack#Affinities (accessed on 4 July 2023).
56. Intent Spoofing. Available online: https://docs.ostorlab.co/kb/INTENT_SPOOFING/ (accessed on 4 July 2023).
57. CVE-2017-13156. Available online: https://nvd.nist.gov/vuln/detail/CVE-2017-13156 (accessed on 4 July 2023).
58. CWE-532: Insertion of Sensitive Information into Log File. Available online: https://cwe.mitre.org/data/definitions/532.html (accessed on 4 July 2023).
59. CWE-926: Improper Export of Android Application Components. Available online: https://cwe.mitre.org/data/definitions/926.html (accessed on 4 July 2023).
60. European Commission. Cybersecurity Policies. Available online: https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-policies (accessed on 4 July 2023).