Security Challenges of Financial Mobile Apps in Africa

CyLab-Africa
Carnegie Mellon University Security and Privacy Initiative

UPANZI NETWORK

approov
mobile security

Contents

Abstract

This groundbreaking study, the first of its kind in Africa, focused on a thorough analysis of widely-used applications across the continent. The findings revealed a concerning trend: approximately 272 million users have downloaded apps that inadvertently reveal sensitive, high-risk secret keys. These keys include authentication keys, private keys, code push keys, and payment keys. Consequently, this large user base is at risk of potential cyberattacks due to the exposure of these critical secrets.

The adoption of mobile applications for financial services is experiencing a remarkable surge in Africa, as illustrated by the substantial 39% [1] rise and another 22% [2] rise in mobile money transactions on the continent in recent years. The widespread adoption of mobile banking and payment solutions has provided unparalleled convenience and accessibility to underserved populations, promoting financial inclusion and reducing barriers to economic progress [3]. Notably, around 46% (approx. 613 million unique mobile subscribers) of global mobile money registered users were located on the African continent [3]. As financial services become more digitized and accessible through mobile platforms, the potential risks associated with the exposure of confidential information have escalated. A staggering statistic reveals that nearly one in five successful cyberattacks (18%) is aimed directly at financial institutions [14]. It is imperative to understand and address these security concerns to ensure the safeguarding of both user data and the integrity of financial systems.

In this study, the researchers investigated the prevalence of unsecured secrets in binary packages of financial Android applications used in Africa, where secrets included passwords, Application Programming Interface (API) keys and private keys for cryptographic operations. Numerous guides on security best practices discuss the extraction of sensitive information from code and suggest key management systems to prevent sensitive keys from ending up in version control systems. This study focuses on the digital keys found in reverse engineered Android Application Packages (APKs) which may imply that developers do use key management systems but ultimately the keys find their way into the binary package. 224 of the most popular android based financial applications were sampled from across Africa and downloaded and extensively analyzed in Q3 2023. (Note: The applications analyzed included regional and global applications and were selected based on end-user popularity in the selected regions).

The very nature of these applications, handling sensitive personal and financial data and transactions, necessitates a comprehensive analysis of their security. By delving into the secret keys within the binary package of these applications, we aim to shed light on potential risks if the keys land in a malicious actor's hands. A "secret" in the context of digital signatures stored

> **The very nature of these applications, handling sensitive personal and financial data and transactions, necessitates a comprehensive analysis of their security. By delving into the secret keys within the binary package of these applications, we aim to shed light on potential risks if the keys land in a malicious actor's hands. Secrets are essential for verifying the identity of the application and protecting against unauthorized access, tampering, or data breaches. They should be stored in a secure manner to prevent exposure to potential attackers.**

in mobile applications is a confidential and secure piece of information, such as cryptographic API keys, private keys, or sensitive passwords. They are used to authenticate the application and authorize access to protected resources or services, ensuring the integrity and security of data exchanges between the application and a server. Secrets are essential for verifying the identity of the application and protecting against unauthorized access, tampering, or data breaches. They should be stored in a secure manner to prevent exposure to potential attackers. These secret keys are often present in the compiled source code of these applications and may also be inadvertently published to public repositories like Github.

This study, which meticulously analyzed 224 Android financial applications, provides a comprehensive assessment of the extent to which confidential keys are embedded within the binary package and analyzes the number of people that will be potentially affected if the keys leak.

In the subsequent sections of this report, we explain the research methodology, key findings, compare the United States of America (USA) and Europe to Africa, and offer insights into the implications of our discoveries for the broader landscape of mobile application security and financial services in Africa and beyond.

Introduction

Drawing inspiration from a prior investigation conducted by Approov Mobile Security in the USA and Europe [11], our study seeks to draw comparisons between those regions and Africa, pinpointing trends, commonalities, and disparities pertaining to the exposure of secret keys in mobile application's binary package. These comparative insights could potentially offer valuable guidance for policymakers, developers, and security professionals, aiding in the formulation of targeted strategies to enhance the security posture of financial applications on a global scale.

In the principle of code reuse most applications will use cloud services and APIs. A robust operational framework for these mobile applications hinges on the fundamental requirement that access to cloud services and APIs is limited to genuine users with non-malicious intentions. To fortify this framework, a series of safety measures must be in place to verify the authenticity of users, ensure the utilization of an unaltered mobile application version, guarantee the integrity of the device in use, establish a secure communication channel directly connecting to the API server, and enforce exclusive access to the API solely through authorized means. These facets collectively represent the potential entry points exploited by malicious actors seeking to compromise the system's security of the android applications.

In practice, when mobile apps interact with third-party APIs, the process entails registering and obtaining a unique key (API key). This key serves a dual purpose: it identifies the app to the backend API and validates the legitimacy of the requesting app, thereby establishing a clear link between the requesting entity and the API backend. This mechanism effectively prevents unauthorized or anonymous access attempts and provides a means to regulate the flow of data requests.

In the context of mobile development, the spectrum of secret keys extends beyond API keys to include encryption keys for securing sensitive data, authentication keys for accessing services, and signing keys for verifying data authenticity. Additionally, database credentials, OAuth (Open Authorization) client secrets, push notification keys, code push keys, payment gateway secrets, encryption initialization vectors, license keys, and sensitive configuration settings may also need to be embedded.

The analysis revealed a concerning trend in the industry's approach to handling secret keys in mobile application development. The results emphasize the urgent need to create awareness among developers and security test teams on secret management in version control and

Android applications binary packages.

The Methodology

Identifying Tools

This study employed three open source tools, Gitleaks [4], Trufflehog [5], Apk_api_key_extractor [6], that collectively provide a comprehensive assessment of the extent to which confidential keys are embedded within the binary APK packages.

Identifying Apps

Appfigures [7], a platform renowned for its comprehensive analytics and in-depth reports concerning app store rankings, downloads, revenue, and various vital metrics, was employed to meticulously select the top 10 apps based on revenue from a number of African countries within the five distinct regions in Africa—Southern, Central, Eastern, Western, and Northern Africa. This meticulous approach was undertaken to ensure equitable representation across all these diverse regions.

Within each region, a minimum of three countries were chosen totalling to 19 African countries. These countries included Angola, Benin, Burundi, Botswana, Cameroon, Côte d'Ivoire, Algeria, Egypt, Gabon, Ghana, Kenya, Nigeria, Rwanda, Sierra Leone, Tanzania, Morocco, South Africa, Uganda, and Zimbabwe. Additionally, the chosen apps originated from a broad spectrum of subcategories, including but not limited to mobile banking, payment & money transfer, trading & investment, cryptocurrency, mobile money, personal finance, and government service apps.

Below is a brief description of the subcategories. Each app was assigned a main category based on its main functionalities (i.e what services the app mainly provides) and/or who owns the application:

- **Loans** - A loan app is a mobile application that enables users of smartphones and other mobile devices to apply for and receive loans [8].
- **Mobile Banking/Traditional Banking** is a service that allows customers to access their bank accounts and perform various banking tasks, such as transferring money, checking account balances, and paying bills, using a

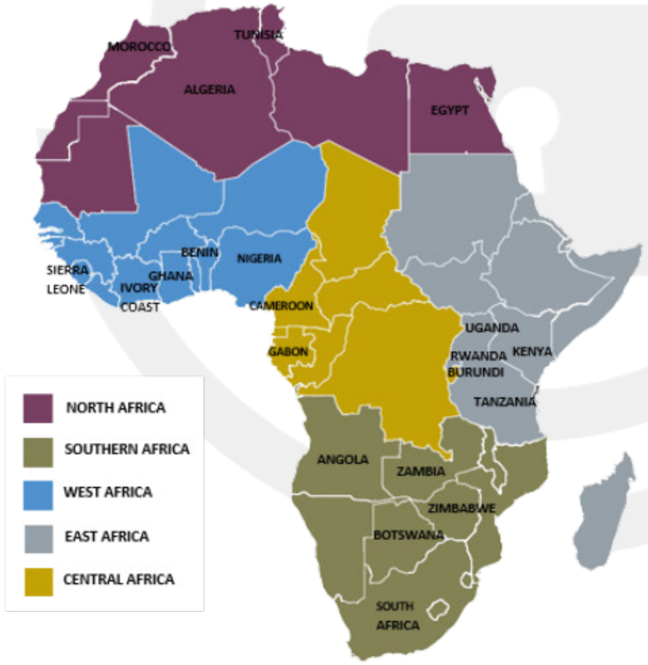
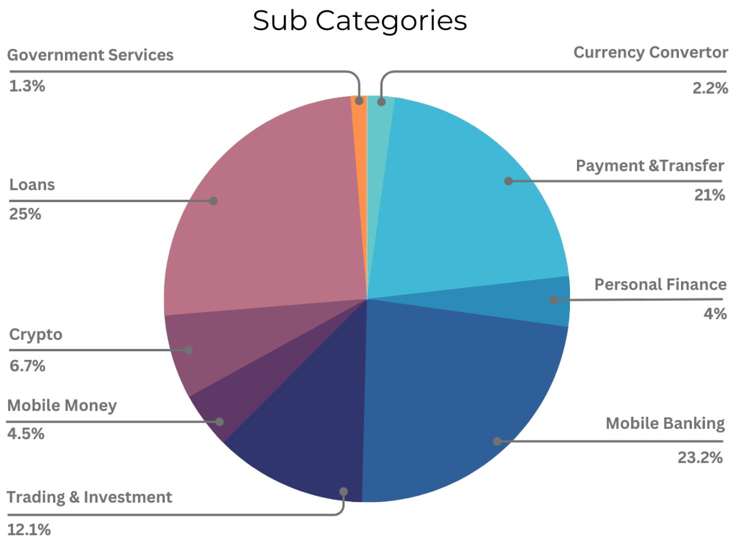
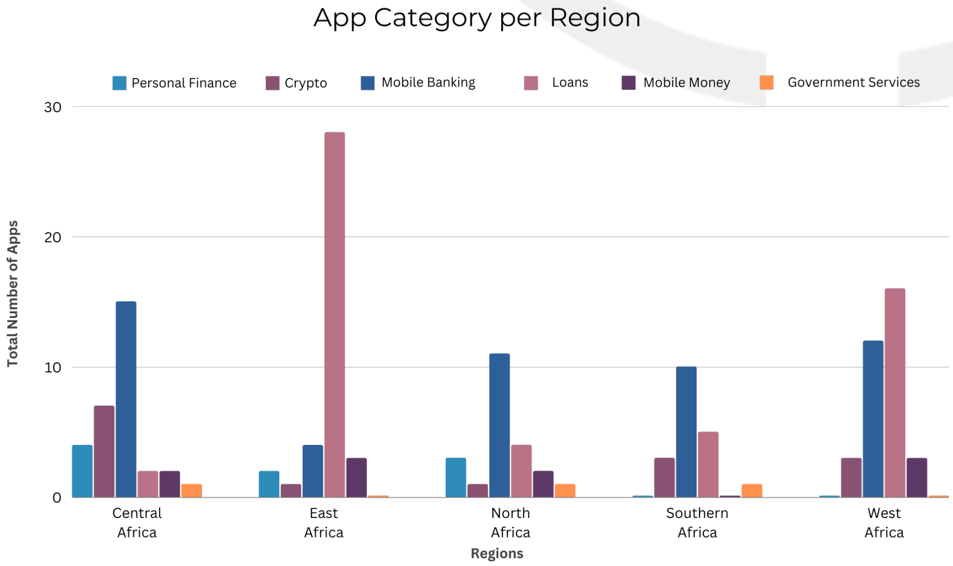


Figure 1: The five distinct African regions and the corresponding countries where a team carried out testing for financial mobile applications. (Source: https://publications.parliament.uk/pa/ld5801/ldselect/ldintrel/88/8806.htm)

mobile device, such as a smartphone or tablet [9].

- **Payment and Transfer** also called peer-to-peer (P2P) money transfer apps, let you transfer cash from person to person, or from entity to entity, quickly, conveniently, cheaply, and securely. They simplify payments and also allow you to use a digital wallet to link your credit card or bank account [10].
- **Trading and Investment** applications give access to trading platforms.
- **Crypto** applications are mainly used for buying, selling and generation of crypto(mining).
- **Mobile Money** involves and is dependent on a phone number and developed and provided by Mobile Network Operator (MNO) and facilitates services ranging from person-to-person transfers, transactions from the bank etc.
- **Personal Finance** - Managing one's finances, budgeting.
- **Currency Converter** apps that provide real-time exchange rates for different currencies.
- **Government Services** provide services to citizens e.g Paying taxes.


Sub Categories


App Category per Region

Out of the total of 224 applications, loan apps stood out as the most prevalent, comprising 56, trailed closely by mobile banking with 52, and payment & transfer with 47. These three subcategories, out of a total of nine, collectively made up approximately 70% of the total mobile financial applications.

Analyzing Apps

After identifying the applications, the team collected each app's ID (appID/packageName) and then used an automated script to download the apk files, reverse engineer (decode) them, scan for secrets, and produce a report with found keys classified as High, Medium and Low Severity.

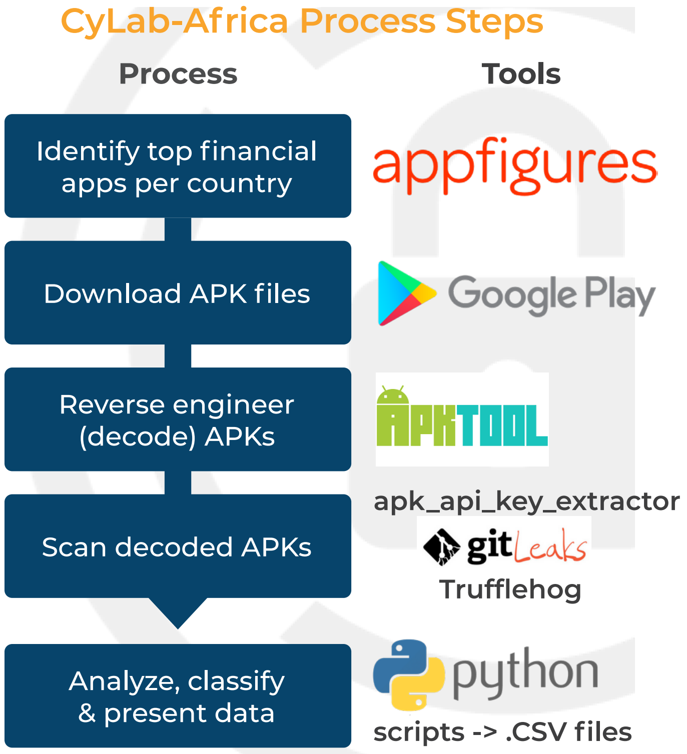
Below is a brief description of the severity risks according to the 2022 Approov report [11]:

High Severity: High Value Secrets are those we consider extremely dangerous if exposed. Some examples: private keys, keys for payment or transfer services and keys that included "authentication" or "attestation". High severity secrets pose a significant risk to the security and integrity of the applications they are embedded within.The exposure of these secrets could potentially lead to unauthorized access, data breaches, and compromised user privacy.

Medium Severity: Medium severity secrets encompass sensitive data that, if exposed, could potentially compromise the confidentiality of user data and application functionality. Although not as critical as the high severity secrets, the compromise of these secrets could still have significant repercussions.

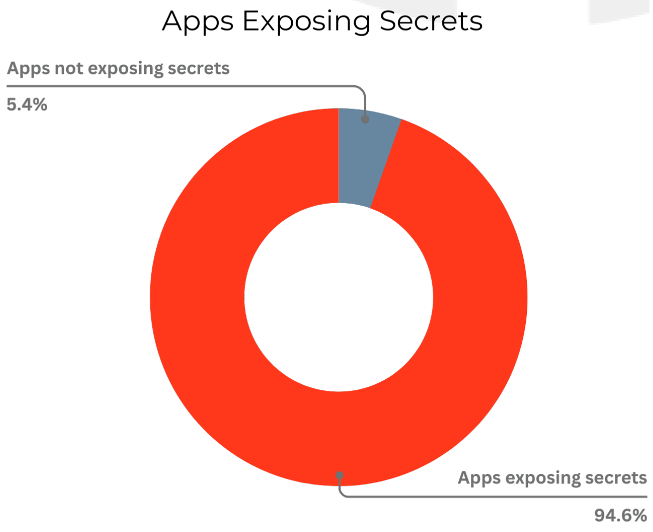
Acceptable (Low) Severity: Low Value Secrets are not considered "service-impacting" e.g. API keys associated with crash or installation analytics. These secrets, while not posing immediate and critical risks to application security, still merit attention from a proactive security perspective. By identifying and categorizing secrets like 'generic_secret_key' we aimed to provide a holistic overview of potential security vulnerabilities that developers should be aware of. While these secrets may not lead to immediate breaches, their exposure could contribute to an accumulation of vulnerabilities that malicious actors might exploit in conjunction with other security weaknesses.

All the apk files were downloaded from the Google Play Store. First the APKs were decoded using the apk_api_key_extractor. This decodes the APK with the bundled apktool and then extracts the secrets and classifies them with a neural network. Lastly, we ran the Gitleaks and Trufflehog scanners against the folder containing all the decoded APKs.



CyLab-Africa Process Steps

Process	Tools
Identify top financial apps per country	appfigures
Download APK files	Google Play
Reverse engineer (decode) APKs	APKTOOL
Scan decoded APKs	apk_api_key_extractor gitLeaks Trufflehog
Analyze, classify & present data	python scripts -> .CSV files



After the scanners completed the extraction of secrets and information, we ran a custom python script to process the findings of each scanner, using a combination of whitelists and blacklists to eliminate false positives and classify the secrets found. And then lastly all the results for each app were consolidated into one csv file for analysis.

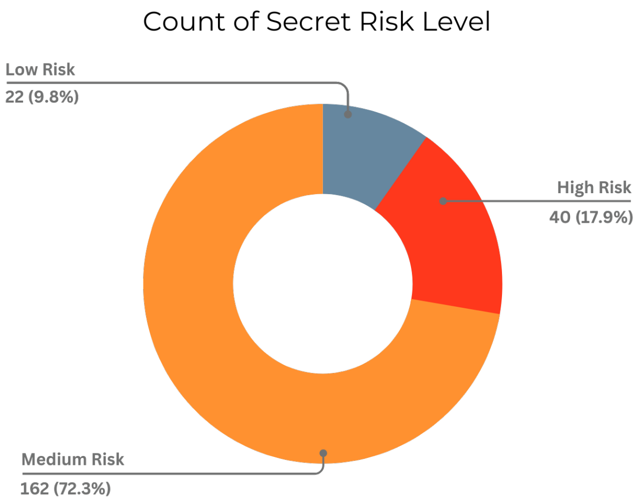
Findings

Apps Exposing keys

Out of the total of 224 applications, a mere 5.4% refrained from revealing secrets, whereas a significant 94.6% did indeed expose secrets.

Secrets Risk Classification

The greater portion (72.3%) revealed keys of medium risk, succeeded by high risk keys accounting for 17.9%, and subsequently, low risk keys at 9.8%. Among apps possessing a high risk level of secrets, their collective real installs amounted to 271,639,867, whereas apps with medium risk keys totaled 902,886,796 installs. Lastly, applications with low risk keys were responsible for a combined total of 44,751,508 real installs.



Category	High Risk %	Medium Risk %	Low Risk %
Mobile Money	0%	70%	30%
Currency Convertor	0%	80%	20%
Government Services App	0%	100%	0%
Crypto Apps	33.3%	53.3%	13.3%
Trading & Investment	18.5%	70.4%	11.1%
Personal Finance	22.2%	66.7%	11.1%
Payment & Transfer	19.1%	76.6%	4.3%
MobileBanking	3.8%	86.5%	9.6%
Loan Apps	7.1%	83.9%	8.9%

Table 1

App Categories Risk Classification

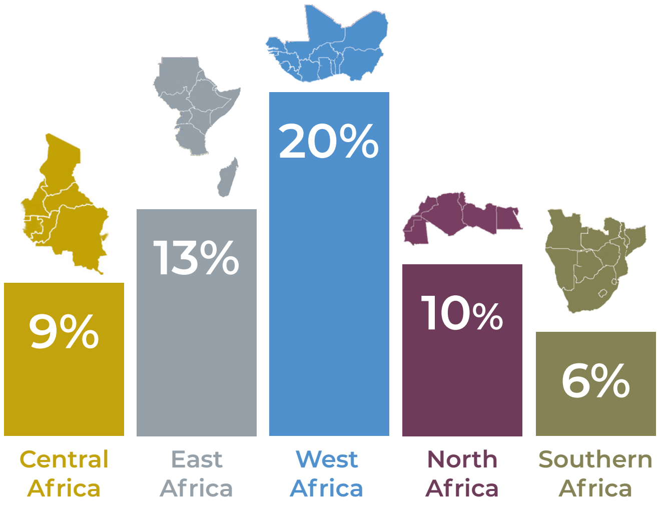
Table 1 illustrates the percentages of applications within different subcategories that revealed high-risk secret keys, medium-risk secret keys, and low-risk secret keys.

App Categories Exposing High Risk Keys

Perhaps not surprisingly, cryptocurrency applications exhibited the highest percentage of high-risk secret keys at 33.3%, followed by Personal Finance applications at 22.2%. Subsequently, Payment & Transfer apps and Trading & Investment apps had lower proportions of 19.1% and 18.5% for high-risk keys, respectively.

Region with Most High Risk Keys

Our research revealed not insignificant regional differences in the percentage of Android applications revealing “high risk” secrets where West and East Africa had marginally higher percentages of 20% and 13% respectively, while Southern, Central and North Africa performed better with 6%, 9% and 10% respectively.



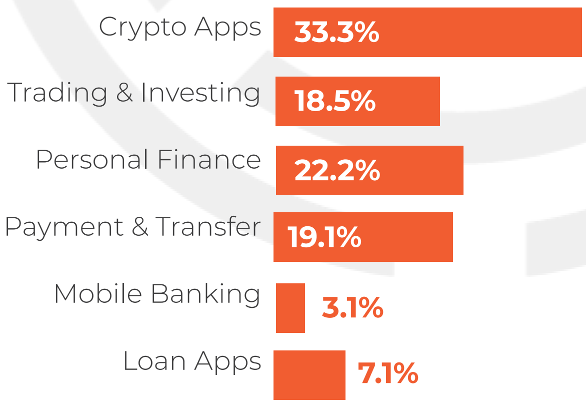
It’s essential to conduct further research to pinpoint the specific factors contributing to these regional differences in secret exposure. By understanding the underlying causes, it becomes possible to develop targeted strategies to enhance the security of Android applications across Africa.

App Categories Exposing Medium Risk Keys Overview

Concerning medium-risk keys, every subcategory had over 50% of its applications revealing these medium-risk keys. Government services, Mobile banking, Loan Apps, and Payment & Transfer apps all had more than 75% of their applications exposing medium-risk keys.

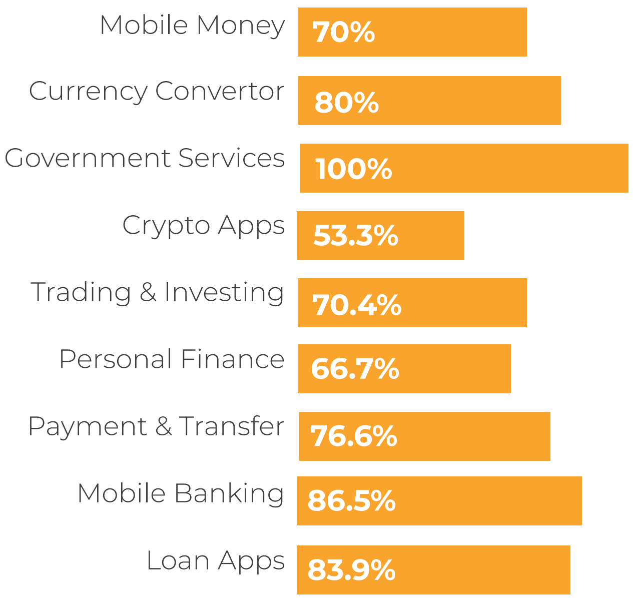
The percentage of apps leaking dangerous secrets and exposure was consistently high — around 20% for the main app categories. Crypto had the highest percentage at 33.3%.

Categories with High Risk Secrets



Government services, Mobile banking, Loan Apps, and Payment & Transfer apps all had more than 75% of their applications exposing medium-risk keys.

Categories with Medium Risk Secrets



Type of Secrets Found

Table 2 illustrates the types of discovered keys, highlighting that the predominant secrets were cloud provider keys. These predominantly encompassed google_api_keys and a smaller number of gcp_api_keys. Following closely were social network keys, involving facebook_oauth and facebook_client_token. Subsequently, deployment keys for code deployment and authentication keys, comprising social network authentication and an array of OAuth tokens, were observed. Additionally, there were fewer occurrences of other keys such as private-keys,

High Risk Secrets

Type of Secret Key	Number of Apps with the Secret
Private Keys	9
Mobile Attestation	6
Payments	13
Encryption Keys	3
Authentication	6
Deployment	12

Medium Risk Secrets

Type of Secret Key	Number of Apps with the Secret
Messaging & Notifications	15
Cloud Providers	194
Analytics & User Tracking	79
Social Networks	36

Low Risk Secrets

Type of Secret Key	Number of Apps with the Secret
Generic	19

Table 2

cryptography keys, and payment API keys.

Notably, Google cloud API keys were identified in 86% of the examined applications, while Facebook client tokens were present in 12% of them. Furthermore, approximately 15.3% of the apps exposed various authentication tokens, including Facebook authentication tokens.

Key Takeaways

West Africa has the highest percentage of high risk secrets exposed within Africa. This may be due to the fact that generally crypto apps and investment & trading apps had a high percentage of high risk keys, and more of them are being used in the Western region in comparison to other regions.

Mobile attestation is commonly used in scenarios like mobile banking, mobile payment systems, and enterprise mobile applications, where ensuring the security of both the device and the application is critical. It helps protect against various threats, including malware, device compromise, and unauthorized access to sensitive data. The Google Play SafetyNet attestation API by Google was used by 2 apps out of 224 tested and none of the applications employed Google Play Integrity API despite the fact that the Google attestation API will be deprecated in December 2023.

Crypto apps have a high percentage of high risk secret keys

The reason behind the frequent leakage of high-risk secrets and API keys in crypto apps can be traced to the intricate nature of blockchain-based software development and the vulnerabilities inherent in decentralized finance (DeFi) architecture [13]. Unlike traditional applications, the continuous integration and deployment on blockchains create an environment where even a small coding error can lead to significant security breaches. While blockchain technology is often seen as secure, malicious actors have found ways to exploit vulnerabilities, expanding the attack surface within DeFi environments. Additionally, as network boundaries blur and workloads scale on blockchains, beyond an organization’s control, the risk of introducing or exploiting vulnerabilities in the codebase becomes more pronounced. Traditional banking and payment apps are often subject to stringent financial regulations, which can require robust security measures. Crypto apps may not have the same level of regulatory oversight, leaving them more vulnerable. To address this issue, crypto app developers must adopt secure coding practices, consult blockchain-specific security documentation, and utilize third-party tools for rigorous security validation.

It’s also worth noting that security vulnerabilities in crypto apps do not in any way imply that traditional banking and payment apps were immune to risks, as they also exposed secrets. The relatively new and rapidly evolving nature of cryptocurrency and the global, decentralized, and less-regulated ecosystem surrounding it can make crypto apps more susceptible to security issues. As the crypto industry matures, it’s expected that security practices may improve, but vigilance is essential in the interim and with all financial apps in general.

Crypto Apps also had the highest number of high severity secret keys in both Africa and Western Countries, including the United Kingdom, France, Germany and the United States.

Comparing Security Challenges in Africa to the United States and Europe

Africa is facing similar challenges with financial applications as other parts of the world. In comparing the exposure of secrets, the results were remarkably similar and indicative of a global challenge more than a regional problem:

- The same types of exposed keys were found on all three continents (Africa, Europe, N. America).
- The USA and Europe had 93 % exposing secrets while Africa had 95% exposing secrets.
- The USA and Europe had 23% exposing **high** secrets while Africa had 17.9% exposing **high** secrets.
- Only 6 out of the 224 applications tested in Africa had some form of mobile attestation and this trend was also exhibited in the USA and Europe.

Based on these results it is inconclusive as to which parts of the world are actually more vulnerable to cyber attacks and potential fraud. However, when comparing security challenges in Africa to those in the United States and Europe, several additional factors may come into play:

- **Infrastructure:** Developed regions like the United States and Europe generally have more advanced and robust technology infrastructure, including secure networks and data centers. In contrast, some parts of Africa may still face challenges in terms of infrastructure development, which could impact the security of financial mobile apps.
- **Regulatory Framework:** The regulatory frameworks for data protection and cybersecurity may differ across regions. The United States and Europe have established regulations such as the General Data Protection Regulation (GDPR) in the European Union and various data protection laws in the United States. These regulations aim to protect user data and ensure privacy. In Africa, regulatory frameworks for data protection and cybersecurity are still evolving.
- **Awareness and Education:** The level of awareness and education regarding cybersecurity practices can vary between regions. Developed regions often have higher levels of awareness among both developers and end-users regarding best practices for securing mobile apps and protecting sensitive data. In Africa, there may be a need for increased awareness and education on cybersecurity measures among developers, organizations, and users.
- **Threat Landscape:** The types of cyber threats and attack vectors may differ based on the region. While financial mobile apps face similar risks globally, such as data breaches, account takeovers, and malware attacks, the specific tactics used by cybercriminals can vary. The threat landscape in Africa may involve unique challenges and targeted attacks based on the region's specific circumstances.

It's important to note that the aforementioned points are general observations, and a comprehensive analysis of the security challenges in Africa compared to the United States and Europe would require more detailed research and data.

Implications

Exposing these sensitive keys can have profound and far-reaching implications to developers, users and both the security and functionality of an application. Firstly, the exposure of private keys and encryption keys can compromise the confidentiality and integrity of user data, potentially leading to data breaches and privacy violations. Secondly, OAuth tokens, API keys, and authentication keys are often used to establish trust between the app and external services. When exposed, they can be abused to gain unauthorized access to user accounts or manipulate the application's behavior. Thirdly, code push keys, when misused, can allow attackers to inject malicious code updates into the app, undermining its reliability and safety. Exposing API keys, especially those related to services like Google, AWS, and other cloud services, can result in unauthorized usage, which may incur unexpected costs or disrupt the functionality of integrated features. In summary, the consequences of exposing these keys are dire, ranging from financial and reputation damage to potential legal repercussions, highlighting the critical importance of robust security practices in Android app development.

Recommendations

Developers

For Android developers building mobile apps for financial services, there are several critical security recommendations to follow. Developers can make use of encryption and hashing as it changes the secret keys into an unreadable format. This can deter attackers due to the difficult nature of decoding or decrypting the unreadable data. It is important that developers implement these security mechanisms using randomized initial vectors (no hardcording) and well-established encryption and hashing methods such as AES or RSA, both locally and during transit. This mechanism is not foolproof as attackers can find ways to intercept traffic at the moment the data is decrypted for use.

Another mechanism to minimize exposing APIs keys and reduce the attack surface is by using secure key management services like Android Keystore system which is good for keeping keys protected at rest on the device and out of the app package, but doesn't help during traffic transit as well as determining authenticity of app/device; use of Native Development Kits (NDK) which also makes it harder to scan for keys in the app package using static methods; and environment variables in the app-level build.gradle file which help keep the secret keys out of the source code.

It's essential to implement certificate pinning (SSL pinning) to protect against manipulator-in-the-middle (MitM) attacks when transmitting sensitive financial data. According to a recent report "Evaluating Mobile Banking Application Security" [12] many apps do not implement this security mechanism. Developers can successfully hide secrets from statics analysis but attackers can also manage to extract these secrets during run time through MitM, thus SSL pinning makes it difficult for attackers to implement MitM attacks. Also, it can be possible to extract these keys with pinning in place but SSL pinning is a step on implementing defense in depth on the app package. To complement SSL pinning on the app packages, developers can also implement runtime protection such as mobile attestation which is a more comprehensive approach to runtime mobile application security. It protects the app, the client environment and the communications channel from the app to the back-end. It can check if requests to the API are genuine , it can check if the app was repackaged, it can detect unsafe environments on the client device, such as rooted/jailbroken devices, it can check if the apps are running under debuggers or emulators.

It is evident that no single method is foolproof, hence implementing a number of recommended security mechanisms increases the security of the application. Regular penetration testing should be conducted to uncover logical vulnerabilities. Advanced obfuscation of the code will make reverse engineering more challenging, and adopting biometric authentication methods like fingerprint or facial recognition as a second factor of authentication enhances user authentication security. Furthermore, developers should securely erase cached or temporary financial data and limit data retention to the minimum necessary. Automated security patching mechanisms should be utilized, and a secure design process that includes threat modeling and risk assessments during the design phase will result in more robust, secure financial apps.

General Public

On the end-user side of the security measures, there are essential recommendations to protect assets and reduce the risk of online fraud when using mobile apps. First and foremost, users should download apps only from reputable stores (e.g Apple Store and Google Play Store) to avoid malicious software. Strong passwords and enabling multi-factor authentication (MFA) wherever possible can greatly enhance account security. Users should exercise caution when connecting to public Wi-Fi networks, as they can be vulnerable to data interception; using VPNs to encrypt traffic is advisable. Keeping apps and the mobile operating system up-to-date is essential as updates often patch known vulnerabilities. When installing apps, users should carefully review and restrict permissions, only granting access to necessary data and device features. Being vigilant against phishing scams is crucial, as cybercriminals may send fraudulent messages pretending to be from financial institutions. Limiting app permissions through OS settings, logging out fully after sessions, and using device locking features with short timeouts and biometric unlocking add additional layers of security. Finally, monitoring financial accounts frequently helps detect and respond quickly to any unauthorized access, protecting one's finances and assets from cybercriminals.

Seizing the African Growth Opportunity: The Imperative for Mobile App Security

Africa is poised for a significant economic upswing over the next decade, with the promise of significant economic growth. As the continent embraces transformation, there's an essential element in the security and resilience of its financial technologies and infrastructure. Digital inclusion is a key driver of this growth and making the security of mobile applications a pressing concern. Fintech innovations and the growing ubiquity of mobile apps for financial services demand a corresponding commitment to the protection of sensitive personal and financial data. As the African economy grows, ensuring the security of digital financial transactions becomes paramount.

In this report, we narrowed our focus to address the crucial issue of safeguarding secrets in mobile applications. Our aim was to contribute to the ongoing dialogue about mobile application security, emphasizing the need for proactive measures to protect sensitive financial information and empower individuals to engage securely in digital financial transactions.

- **Supporting Sustainable Growth:** As African economies expand, security measures must expand in parallel. The sustainability of this growth hinges on creating a secure environment for digital financial transactions.
- **Closing Security Gaps:** Disparities in mobile app security must be addressed. While some regions might exhibit a higher prevalence of vulnerabilities, ensuring uniformly high security standards is essential.

- **Ongoing Vigilance:** Security challenges are ever-evolving. Continuous research is vital in staying ahead of new vulnerabilities and threats that may arise in the fast-paced tech landscape.
- **Investing in Security:** Prioritizing security is an investment in the future. Allocating resources to implement best practices, promote secure coding, and educate developers about security is essential.
- **Empowering Users:** Educating users about security practices is just as critical as app development. Empowering individuals to protect their data is an integral aspect of the security ecosystem.
- **Regional Collaboration:** Collaboration on security standards and best practices can help strengthen security measures across the continent. Regional organizations can facilitate this collaboration to address shared security challenges.

In light of the forthcoming economic growth, we must recognize the paramount importance of securing secrets within mobile applications. By addressing this need for enhanced security, we can ensure that Africa's digital transformation is not only extensive but also safe, inclusive, and prosperous.

.

References

[1] Puskas, P. (2023, March 23). Internationalising mobile banking in Africa — tradeexperettes. TradeExperettes. https://www.tradeexperettes.org/blog/articles/internationalising-mobile-banking-in-africa

[2} Gilbert, P. (2023, April 20). Mobile money transactions hit $1.26T in 2022 – GSMA - Connecting Africa. Connecting Africa. https://www.connectingafrica.com/author.asp?section_id=761&amp;doc_id=784458#close-modal

[3] DocSend - Simple, intelligent, modern content sending. (2019, November). DocSend. https://ftpartners.docsend.com/view/kg7dbcj

[4] GitHub - gitleaks/gitleaks: Protect and discover secrets using Gitleaks (n.d.). GitHub. https://github.com/gitleaks/gitleaks

[5] GitHub - trufflesecurity/trufflehog: Find and verify credentials. (n.d.). GitHub. https://github.com/trufflesecurity/trufflehog

[6] GitHub - alessandrodd/apk_api_key_extractor: Automatically extracts API Keys from APK files. (n.d.-b). GitHub. https://github.com/alessandrodd/apk_api_key_extractor

[7] Analytics, ASO Tools, and App Intelligence by Appfigures. (n.d.). Appfigures. https://appfigures.com/

[8] Juyal, P. (2021). What is a Mobile Banking Application? Know Everything About it!. Retrieved 4 September 2023, from https://paytm.com/blog/net-banking/what-is-mobile-banking-application/

[9] Insights on Loan Apps: What to Know Before Applying for a Loan. (2023). Retrieved 4 September 2023, from https://www.linkedin.com/pulse/insights-loan-apps-what-know-before-applying-iyanuoluwa-oluwatayo/

[10] Brock, M. (2021, April 26). The best money transfer apps. Investopedia. https://www.investopedia.com/best-money-transfer-apps-5180183#

[11] Secret report. (n.d.). Mobile App Protection | Mobile API Security | Approov. https://approov.io/info/secret-report

[12] Trevor Henry Chiboora, Lenah Chacha, Theoneste Byagutangaza, and Assane Gueye. 2023. Evaluating Mobile Banking Application Security Posture Using the OWASP's MASVS Framework. In Proceedings of the 6th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS '23). Association for Computing Machinery, New York, NY, USA, 99–106. https://doi.org/10.1145/3588001.3609367

[13] K. Steinkamp, "Crypto vulnerability management," Coalfire.com, https://www.coalfire.com/the-coalfire-blog/crypto-vulnerability-management

[14] Positive Technologies, "Cybersecurity Threatscape of African countries 2022–2023," ptsecurity.com, https://www.ptsecurity.com/ww-en/analytics/africa-cybersecurity-threatscape-2022-2023/

Future Studies

In the realm of future research, one exciting avenue to explore is the dynamic analysis of how financial Android apps handle secret keys. Imagine actively monitoring how these apps behave during runtime. By doing so, we can uncover how secret keys are managed in practice and whether they're ever exposed during actual operation, revealing potential vulnerabilities that static analyses might miss.

Another intriguing direction involves comparing how Android-based financial apps in Africa handle secret keys compared to their iOS counterparts. Such a study could reveal platform-specific security challenges and solutions, giving us a broader perspective on mobile app security.

Understanding user behavior and security awareness is crucial. A study could delve into how conscious users are of security practices like password management and how these behaviors affect the overall safety of their financial data. This insight into user perceptions could be invaluable for tailoring security measures effectively.

Furthermore, we could conduct in-depth case studies on security breaches in African financial Android apps. By analyzing specific incidents, we can uncover their root causes and assess the real-world impact on users, providing invaluable insights into actual security challenges.

Additionally, we could establish comprehensive best practices for secure key management in these apps, covering encryption, storage, and access control. Such guidelines would serve as a practical framework for developers to enhance security.

Moreover, compliance and regulatory assessments could gauge how well these apps align with local and international security standards. These evaluations would shed light on areas needing improvement to ensure legal and security compliance.

Enhancing user education and training programs could be pivotal. A study could assess the effectiveness of such initiatives in empowering users to adopt secure practices and safeguard their financial data.

For developers, customized secure development training programs tailored to the African context could be beneficial. We could explore whether such training leads to improved coding practices and fewer vulnerabilities in financial Android apps.

Lastly, continuous monitoring of the evolving threat landscape surrounding these apps in Africa is crucial. Such ongoing research would help us identify emerging threats and adapt security measures accordingly in this rapidly changing environment.

Authors

Theoneste Byagutangaza Research Associate, CyLab-Africa

Theoneste Byagutangaza's expertise lies in application and network security, focusing on enhancing the safety of financial technology and health applications. Byagutangaza excels in penetration testing, network security, and threat intelligence analysis. His experience is built on rigorous research and hands-on work, positioning him as a trusted guardian against evolving cyber threats.

Before joining CyLab-Africa as a research associate, Byagutangaza served as a senior network security engineer at Alpha Computer, extending his expertise to banks in Rwanda and Sierra Leone, government entities, and health organizations in Rwanda. His dedication was evident in strengthening cyber defenses and safeguarding critical systems.

Byagutangaza holds a master's degree in information technology with a focus in cybersecurity from Carnegie Mellon University Africa. He began his academic journey at the University of Rwanda, where he earned his bachelor of science in information technology.

Trevor Henry Chiboora Research Associate, CyLab-Africa

Trevor Henry Chiboora currently serves as a research associate at CyLab-Africa, where he works on Vulnerability Assessment and Penetration Testing project. In this role, he conducts research related to cybersecurity and specializes in performing pentests on mobile and web applications. Notably, he played a key role in deploying a state-of-the-art Security Operations Centerusing Elastic Security, MISP, and other integrations.

Chiboora's earned a bachelor of technology in information technology with a first-class degree (distinction) from the Harare Institute of Technology. He has a master of science in information technology degree with a specialization in cybersecurity and computer networking from Carnegie Mellon University Africa. Chiboora also has earned certifications including the Cisco Certified CyberOps Associate and Certified in Cybersecurity (CC). Chiboora's expertise include: network troubleshooting and configurations, network and application security, SOC analysis and endpoint security.

Joel Jefferson Musiime Research Assistant, CyLab-Africa

Joel Jefferson Musiime is a 2nd year graduate student pursuing a master of science in information technology at Carnegie Mellon University Africa. He also serves as a part-time research assistant in CyLab-Africa, where he works on the Vulnerability Assessment and Penetration Testing team. In this role, he conducts security research and assessments on web and mobile applications. He earned a bachelor of information technology and computing from Kyambogo University, Uganda and holds a CompTIA Security+ certification. He also has previous experience working as a software developer designing ERP solutions for businesses. Musiime's areas of interest include mobile application security, digital privacy, and API and cloud security.

Lenah Chacha Research Lab Manager, CyLab-Africa

Prior to joining CyLab-Africa, Lenah Chacha has held various roles in networking, security engineering, application security engineering, and instructor.

She has also been a research assistant at the lab working on our flagship projects like MOSIP, VAPTs and picoCTF. She has consulted on various levels as picoCTF PI, project lead for industry VAPTs and surveys and team lead for security analysis and investigations for the lab's MOSIP deployment. She joins Cylab-Africa with more than seven years of consultancy on security where she has guided organizations design and finetune their security strategy.

Chacha received her graduate degree from Carnegie Mellon University Africa and is also a certified Ethical Hacker.

Advisors

Assane Gueye Associate Teaching Professor, Carnegie Mellon University Africa, Co-director, CyLab-Africa, Co-director, Upanzi Open Digital Technologies Network

Prior to joining CMU-Africa, Assane Gueye was a faculty member at the ICT Department at the University Alioune Diop of Bambey, Senegal, where he also leads the research group "Technologies de l'Information et de la Communication pour le Développement" (TIC4Dev). Gueye also holds a guest researcher position with the National Institute for Standards and Technology, Gaithersburg, Maryland, USA.

Assane completed his Ph.D. in electrical engineering and computer sciences from UC Berkeley in March 2011. He received a master's degree in 2004 in communication systems engineering from Ecole Polytechnique Fédérale de Lausanne, Switzerland.

His research focuses in two main areas: performance evaluation and security of large-scale communication systems, and information and communication technologies for development (ICT4D). Assane is a Fellow of the Next Einstein Forum (Class of 2016). In 2019 he was nominated as a member of the European Alliance for Innovation (EAI) inaugural Fellow Class.

Giulia Fanti Assistant Professor of Electrical and Computer Engineering, Carnegie Mellon University, Co-Director, CyLab-Africa

Giulia Fanti's research interests span the algorithmic foundations of blockchains, distributed systems, privacy-preserving technologies, and machine learning. She is a fellow for the World Economic Forum's Global Future Council on Cybersecurity, and has received a best paper award at ACM Sigmetrics and an NSF Graduate Research Fellowship. She obtained her Ph.D. in EECS from U.C. Berkeley and her B.S. in ECE from Olin College of Engineering.

Ted Miracco CEO, Approov Limited

Ted's high-technology experience spans 30 years in cybersecurity, electronic design automation (EDA), RF/microwave circuit design, semiconductors, and defense electronics. Prior to his role as CEO of Approov Mobile Security, Ted co-founded Cylynt (formerly SmartFlow) in 2014 and served as the Chief Executive Officer until August 2022. Under his leadership, Cylynt experienced significant growth and became the leading anti-piracy solution provider in various software industries. He obtained his B.S. in ECE from Carnegie Mellon University.

Richard Taylor CTO, Approov Limited

Richard has more than 30 years industry experience, with a background compiler optimization and processor architecture, working more recently in application security and cloud computing technologies. Richard co-founded Approov Mobile Security and has led a number of innovative product developments in the areas of EDA, software optimization and remote software attestation. He is passionate about solving complex technical issues and holds a number of patents.

The CyLab-Africa initiative is a collaboration between Carnegie Mellon University's CyLab Security and Privacy Institute and Carnegie Mellon University Africa. The initiative aims to improve the cybersecurity of digital systems in Africa and other emerging economies. CMU-Africa, located in Kigali, Rwanda, is the only U.S. research university offering its master's degrees with a full-time faculty, staff, and operations in Africa. The institution, part of Carnegie Mellon's College of Engineering, is addressing the critical shortage of high-quality engineering talent required to accelerate the economic transformation of the African continent. Find out more: https://www.africa.engineering.cmu.edu/research/cylab/index.html

Approov is considered a cornerstone of mobile application security for leading global organizations whose consumer and B2B applications are used by millions annually, including eCommerce, financial services, healthcare, gaming and connected car sector organizations. Approov provides a comprehensive runtime security solution for mobile apps and their APIs, unified across iOS and Android. Find out more: https://approov.io/