Forensic Analysis of Mobile Banking Apps

Oluwafemi Osho[1], Uthman L. Mohammed[1], Nanfa N. Nimzing[1], Andrew A. Uduimoh[1], and Sanjay Misra[2](✉)

[1] Federal University of Technology, Minna, Nigeria
{femi.osho,a.uduimoh}@futminna.edu.ng
[2] Covenant University, Ota, Nigeria
sanjay.misra@covenantuniversity.edu.ng

Abstract. Over the years, the proliferation of mobile banking applications has been on the increase. Financial institutions are taking advantage of mobile technology to provide accessible, ubiquitous, user-friendly, convenient, and cost-effective services to their customers. The mobile banking applications access and process sensitive user data. As such, they are required to manage such data in a high secure manner and run in secure environment. This study conducts a forensic investigation of twelve popular Android m-banking apps in Nigeria to determine if the generated backups by the mobile OS do not save sensitive data; the application removes sensitive data from view when backgrounded; sensitive data are not held longer than necessary in the memory, with the memory cleared after use; minimum device access security policies are enforced by the app, and users are educated by the app about the type of PII processed and security best practices in using the app. Our findings revealed that while none of the apps saved sensitive data in generated backup, all except one held data of sensitive value in the memory of the test device and did not enforce any device access security policy. Also, none of the apps removed sensitive data when backgrounded. In addition to serving as a source of information for forensic investigators, we believe our study could assist mobile banking app developers in identifying aspects of the development process that need attention, which would lead to better secured apps.

Keywords: m-banking · Forensic · UFED · FRED

1 Introduction

Globally, there has been a constant increase in the adoption of mobile devices [1]. A forecast by Statista [2] estimated a growth in the number of smartphone users from 2.1 billion in 2016 to 2.5 billion in 2019.

With improvement in the processing power of smartphones, relatively at par with computers, and array of functionalities provided, more banks continue to take advantage of mobile technology in their quest to offer personalized and customer-oriented financial and non-financial services to their customers, in ways that are more ubiquitous, accessible, user-friendly, convenient, and cost-effective [3–8].

S. Misra et al. (Eds.): ICCSA 2019, LNCS 11623, pp. 613–626, 2019.
https://doi.org/10.1007/978-3-030-24308-1_49

Mobile banking, also known as m-banking, is growing in popularity. In the US, m-banking apps are one of the top three most used apps [9]. It has been reported that by 2021, over 2 billion people will have used their mobile devices for banking [10]. Reports have also shown that more bank customers are choosing it over e-banking [10, 11]. While the common activity is checking account balance, users also engage m-banking apps for paying bills and transferring money to other people.

The situation in Nigeria is no different from those in most of the other countries. There has been dramatic increase in mobile usage [12, 13]. From around 110 million mobile subscribers in 2012, the number of mobile users by December 2018 had grown by more than 120% to above 250 million [14]. This has resulted in the proliferation of mobile banking services in the country, which has contributed significantly towards the implementation of cashless economy in the country [15].

However the benefits that mobile banking offers, studies have identified security risk as one of the main factors that negatively impact its adoption [7, 16]. At the core of any m-banking app is security [3]. The fact is, attackers are less likely to gain physical access to web servers than to mobile devices. The implication is that data on memory of mobile devices could be more susceptible to unauthorized access by attackers than those on web servers [17]. Regrettably, compared to other devices, one disadvantage associated with mobile devices is increased likelihood of being stolen or lost. An attacker who lays hold of such device could gain access to sensitive data. It has been reported that attacks against mobile devices have grown in number and sophistication [18]. This underscores a need for security of these data.

The OWASP's Mobile AppSec Verification Standard (MASVS) stipulates two security verification levels: L1 and L2 [19]. The MASVS-L1 defines some sets of mobile app security best practices. On the other hand, the MASVS-L2 consists of advanced security controls beyond the standard requirements. Mobile banking apps were categorized under MASVS-L2. With regards to data storage and privacy, seven security verification requirements are stipulated for L1. For a mobile app to achieve MASVS-L2, five additional requirements must be satisfied. These include: (1) Generated backups by the mobile OS do not save sensitive data, (2) When backgrounded, the application removes sensitive data from view, (3) Sensitive data are not held longer than necessary in the memory, with the memory cleared after use, (4) Minimum device access security policies are enforced by the app, and (5) Users are educated by the app about the type of PII processed and security best practices in using the app.

Very few studies have focused on forensic analysis of mobile banking apps [20]. Fewer works have investigated Nigerian mobile banking apps. Our study therefore seeks to investigate twelve of the most popular mobile banking apps in Nigeria based on the five MASVS-L2 additional requirements.

The findings in this research will serve as a source of information for forensic investigators. It will assist mobile banking app developers in identifying aspects of the development process that need attention, which would lead to better secured apps. For users of m-banking apps, the study will not only serve as an awareness tool, but also could incentivize them to take the security of their mobile devices more seriously. For instance, being aware that PII are stored in memory for long should naturally motivate a user to be more security-conscious.

The rest of the study is organized as follows: section two summarizes related studies. In section three, the experiment setup is discussed. The findings are presented in section four. The study concludes in section five.

2 Related Studies

Many studies have been conducted in the area of forensic extraction of evidentiary artifacts in mobile devices. While many have focused on Android-based devices, some considered other operating systems, such as Windows and iOS. While some studies analyzed the devices, without focusing on any particular app, e.g. [18], in most literature, specific apps were considered.

One of the mostly covered were social networking apps. In the work of Al Mutawa et al. [21], three social networking apps: Facebook, Twitter, and MySpace were analyzed. Each was installed on Android, Blackberry, and iPhone devices. Analysis of acquired logical images revealed substantial amount of evidentiary data extracted from the Android and iPhone devices, while none was retrievable from the Blackberry device. Another study by Alyahya and Kausar [22] investigated data stored by Snapchat application on an Android device, Samsung Galaxy Note GT-N7000, using Autopsy and AXIOM Examine. Both forensic tools extracted different amount of data. However, one of the issues with AXIOM, the authors reported, was that deleted snaps could not be presented. Autopsy, on the other hand, could not preview databases and indicate senders and receivers of snaps.

Another category of apps were instant messaging apps. Walnycky et al. [23] analyzed 20 popular instant messaging apps for evidentiary data. In most of the apps, data such as passwords, pictures, audios, videos, and more were either intercepted or reconstructed. In [24], a forensic analysis of Kik messenger on Android devices was performed. Artefacts extracted included deleted contacts, messages from deleted contacts, deleted chats and exchanged files. Ovens and Morison [25] also analyzed the Kik messenger app, however on iOS device. They were able to extract deleted images, not only from the device, but also downloaded from the kik servers.

Some literature experimented on multiple apps. For instance, Azfar et al. [26] logically analyzed Android phone images on 30 instant messaging (IM), Voice-over IP (VoIP), and Argumentative and Alternative Communication (AAC) apps using XRY. Based on their findings, they proposed a forensic taxonomy for existing communication apps.

Another study that proposed a taxonomy based on evidentiary artifacts extracted from examined apps is [27]. Focusing on mobile health applications, the authors analyzed 40 mHealth apps. Data extracted include user credentials (e.g. login password and PIN), email addresses, and sequence of user locations and food habits.

A thorough search through literatures revealed very few works have been devoted to mobile banking apps. Three of the studies we found actually focused on identifying vulnerabilities on and potential attacks against m-banking apps. Jung et al. [28], in their study, forged seven m-banking apps in Korea, to explore the possibility of exploiting repackaging attack to transfer money to unintended recipients. They found that existing security measures to mitigate this were not effective. Bojjagani and Sastry [29] proposed STAMBA, a security testing framework for Android-based mobile banking apps. The framework was tested on several m-banking apps using four testing mechanisms: static and dynamic analyses, web app server security, and device forensic. These were considered on three levels of security testing: app, communication, and device levels. Their findings revealed 356 vulnerabilities that could be exploited. Another study by Chen et al. [30] performed automated security risk assessment to identify security weaknesses in mobile banking apps. Their research considered the most number of apps examined in any related studies. Proposing an assessment system that combines static program analysis of data and control flows and natural language processing, they tested 693 m-banking apps from more than 80 countries. They found, among other things, a total of 2,157 weaknesses exploitable by attackers.

One of the studies, however, similar in scope to ours, that focused on forensic examination of apps, is that of Chanajitt et al. [20]. The study focused on seven Android mobile banking apps in Thailand. Using two acquisition tools: DD and JTAG, it was discovered that several of the apps did not encrypt user data. Consequently, the authors were able to extract personally identifiable information (PII) such as users' date of birth, PIN code, account number, account type, and account balance.

So far, the only related study that considered m-banking apps in Nigeria is [31]. The authors used UFED Touch and FRED to forensically analyze five m-banking apps. Their investigation focused on identifying sensitive data held in the memory longer than usual and if the data could be used to deduce users' interactions with the apps. Similar to results in other studies, they found PII, such as user login and transaction details, were retained by the apps in the memory of the devices.

Currently in Nigeria, there are up to nineteen banks that provide mobile banking services. It is therefore pertinent to analyze other apps, to ascertain if they manage securely users' sensitive data. Our study, in addition to considering more mobile banking apps, expands the scope of investigation.

3 Experimental Setup

Materials Used

For the test device, we used a Samsung Galaxy SIII SGH-i747 device. The phone runs Android KitKat 4.4.2. Twelve popular mobile banking apps (Table 1) in Nigeria were downloaded and installed. We created user account on each. The registration, authentication, and transaction requirement for each mobile banking application are presented in Table 2. A total of 10 SIM cards were utilized, two of which were used to provide Internet connection. The remaining eight (SIM 1–8) were used in the course of transactions performed. We undertook some transactions, from July 27–August 7, 2017, such as transfer of funds, payment of bills, and recharge of mobile airtime. Table 3 presents transactions performed on the twelve m-banking apps. For acquisition of data from the mobile device, we used the Cellebrite Universal Forensic Evidence Device (UFED) Touch 4.0. To analyze acquired data, we employed the Forensic Recovery Evidence Device (FRED). To ensure that extracted data were handled in a forensically sound manner, we used a removable drive for dumping the memory.

Methods

Data Acquisition Procedures

To extract data from our test device, two acquisition methods were used: manual and physical acquisition.

Manual Acquisition

This method allows us to manually interact with the device [32]. We employed this method to ascertain if data were retained in the internal memory and cache of the mobile device after transactions were performed. To access the device memory, we opened the application manager via Setting > Application manager > All apps. This allows us to confirm any changes in the data size of the internal memory and cache of the device.

Physical Acquisition

Next, we performed a bit-by-bit imaging of the internal memory of our test device using UFED. This was to ensure that access to the lower file systems to extract all necessary data, including deleted ones. The steps followed to physically acquire the memory are presented in Table 4.

Table 1. m-banking apps version and functions

App name	Application functions							
	App version	Fund transfer	Bill payment	Airtime top-up	Open account	ATM/Branch locator	Account statement	Get help
Bank 1	v0.1.3	Yes	Yes	Yes	No	No	Yes	No
Bank 2	v1.4.0.0	Yes	Yes	Yes	No	Yes	No	No
Bank 3	v3.0.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bank 4	v2.3.2	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 5	v1.4.0.0	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 6	v2.2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bank 7	v5.0.0.0	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 8	v1.6.0.0	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 9	v2.3	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 10	v3.0	Yes	Yes	Yes	No	Yes	Yes	No
Bank 11	v5.1.6	Yes	Yes	Yes	No	Yes	Yes	Yes
Bank 12	v2.4.3.22	Yes	Yes	Yes	No	Yes	Yes	Yes

Table 2. m-banking apps registration, authentication and transaction requirement

Source	Registration requirements	Authentication requirements	Transaction requirement
Bank 1	Username, Acct. No, 4-digit PIN, OTP, Password	Username, Password	4-digit PIN
Bank 2	Username, Acct. No, 4-digit PIN, OTP, Password	Username, Password	4-digit PIN
Bank 3	Phone No, Acct. No, 4-digit PIN, OTP, Password	Phone No, Password	4-digit PIN
Bank 4	Phone No, Acct. No., Email address, OTP, Password, Security Question, 4-digit PIN	Phone No, Password	4-digit PIN
Bank 5	Internet banking ID, Acct. No, 4-digit PIN, OTP, Password	Acct. No, Password	OTP
Bank 6	Phone No, Acct. No, 6-digit PIN, OTP ATM card/pin	6-digit PIN	6-digit PIN
Bank 7	Acct. No, Phone number, Internet Banking ID, Password	Username, Password	4-digit PIN
Bank 8	ATM card, Acct. No, Username, Password	Username, Password	4-digit PIN
Bank 9	Phone No, Acct. No, Username, Password, Security Question	Phone No, Password	4-digit PIN
Bank 10	Acct. No, Phone No, Username, Password	Phone No, Password	4-digit PIN
Bank 11	Internet Banking ID, Phone No, Acct. No, Password, Security Question	Username, Password	4-digit PIN
Bank 12	Acct. No, Phone No, Password, Soft token	Acct. No, Password	4-digit PIN

Analysis of Acquired Data

After manual and physical acquisition of the mobile device, we perform both manual and physical analysis of acquired data. The following process, guided by the OWASP Mobile Security Testing Guide (MSTG) [17], were followed to determine how each of the m-banking apps satisfied the five additional MASVS-L2 requirements.

Generated Backups by the Mobile OS Do Not Save Sensitive Data

The FRED was used to analyze the dumped memory of our test device, generated during physical acquisition, to check if sensitive data were present in the auto-back copies of data and settings for the m-banking apps. We followed the process presented in Table 5.

Table 3. Activities performed on the 12 m-banking apps

Transaction date (mm/dd/yy)	Transaction type	Description
07/27/17	Fund transfer	₦3,000 from Bank 4 to Bank 6
		₦5,000 from Bank 2 to Bank 3
		₦4,000 from Bank 6 to Bank 4
		₦6,000 from Bank 1 to Bank 2
		₦5,000 from Bank 3 to Bank 5
		₦5,000 from Stanbic IBTC to Bank 1
07/29/17	Fund transfer	₦3,900 from Keystone to Bank 7
		₦3,140 from Bank 9 to Bank 10
		₦17, 000 from Bank 12 to Bank 7
		₦1,050 from Bank 11 to Bank 12
	Mobile airtime recharge	₦100 on SIM 4 from Bank 10
07/31/17	Fund transfer	₦2,000 from Bank 4 to Bank 12
		₦2000 from Bank 2 to Bank 6
	Mobile airtime recharge	₦200 on SIM 1 from Bank 3
		₦100 on SIM 2 from Bank 5
		₦100 on SIM 2 from Bank 1
		₦100 on SIM 2 from Bank 6
08/03/17	Fund transfer	₦1000 from Bank 1 to Bank 6
		₦1000 from Bank 5 to Bank 2
		₦2000 from Bank 3 to Bank 6
		₦500 from Bank 6 to Bank 4
		₦2000 from Bank 4 to Bank 3
	Mobile airtime recharge	₦200 on SIM 2 from Bank 2
		₦200 on SIM 2 from Bank 4
		₦200 on SIM 2 from Bank 4
08/04/17	Fund transfer	₦1000 from Bank 12 to Access Bank
		₦3,500 from Bank 7 to Bank 9
		₦1,500 from Bank 10 to Bank 11
	Mobile airtime recharge	₦80 on SIM 4 from Bank 8
		₦50 on SIM 4 from Bank 12
		₦150 on SIM 4 from Bank 11
08/05/17	Fund Transfer	₦16,000 from Bank 10 to Bank 12
	Mobile airtime recharge	₦50 on SIM 4 from Bank 7
		₦1000 on SIM 6 from Bank 9
		₦70 on SIM 4 from Bank 11
		₦50 on SIM 4 from Bank 8
		₦100 on SIM 4 from Bank 12

(*continued*)

Table 3. (*continued*)

Transaction date (mm/dd/yy)	Transaction type	Description
08/06/17	Mobile airtime recharge	₦100 on SIM 7 from Bank 7
		₦100 on SIM 4 from Bank 10
		₦100 on SIM 4 from Bank 9
		₦100 on SIM 4 from Bank 11
		₦30 on SIM 4 from Bank 8
		₦20 on SIM 4 from Bank 12
08/07/17	Fund Transfer	₦2000 from Bank 5 to Bank 3
	Mobile airtime recharge	₦200 on SIM 3 from Bank 6
		₦100 on SIM 3 from Bank 3
		₦100 on SIM 2 from Bank 5
		₦100 on SIM 2 from Bank 1
		₦80 on SIM 5 from Bank 7
		₦150 on SIM 4 from Bank 12
		₦120 on SIM 8 from Bank 8
		₦30 on SIM 4 from Bank 11
		₦50 on SIM 4 from Bank 9
		₦65 on SIM 4 from Bank 10
	Bill payment	₦400 electricity bill to PHCN from Bank 4
		₦200 electricity bill to PHCN from Bank 2

Table 4. Physical acquisition procedure

1:	**START** UFED
2:	**BROWSE** to select Samsung GSM SGH-i747 Galaxy SIII
3:	**SELECT** Physical extraction
4:	**SELECT** bootloader option
5:	**SELECT** removable drive, as the destination of the extracted data
6:	**INSERT** removable drive into the USB port of the UFED
7:	**CLICK** continue
8:	**REMOVE** phone battery and reinsert (the phone should remain unpowered)
9:	**CONNECT** Cellebrite extension cable A, with T-133 yellow head, to the phone
10:	**CONNECT** the USB end of the extension cable A to the USB port of the UFED
11:	**CLICK** continue, to initialize the extraction process
12:	**DISCONNECT** phone, once extraction process is completed
13:	**REMOVE** removable drive

Table 5. Physical analysis procedure

1:	**START** FRED
2:	**INSERT** removable drive into FRED workstation
3:	**OPEN** Physical Analyser
4:	**SELECT** Samsung GSM SGH-i747 Galaxy SIII. (The memory dump in.bin format is loaded into the computer memory in clear text)
5:	**OPEN** Analysis page
6:	**OPEN** No_backup folder, for each m-banking app
7:	**ANALYSE** folder contents using Database, Hex View and File Info Format

Application Removes Sensitive Data from View when Backgrounded
Device manufacturers may provide screenshot-saving feature that is used when an application is backgrounded. While an application is displaying sensitive data, these data could be exposed if the application is screenshot. For each app, on a screen that contained sensitive information, such as login page containing login details, we clicked the home button to background the app. We then press the app switcher button to restore the app to the foreground. We observed if the app was screenshot when backgrounded by checking if the screen still contained the sensitive data.

Sensitive Data Are Not Held Longer than Necessary in the Memory, with the Memory Cleared After Use
To determine if sensitive data were only held as briefly as possible in the memory, we followed the same procedure in Table 5, however, instead of the No_backup folder, we checked for the presence of PII, registration- and transaction-related data in the Databases, Cache, Files, Logical storage, Shares_Pref, GPUCache, and APP_Webview folders under each m-banking app.

Minimum Device Access Security Policies Are Enforced by the App
Applications that process and manage sensitive data, to enforce some measure of device access security, can require users to activate some security measures, including setting a device passcode. During registration of each app, after installation, we observed if the app requested us to set a password for the test device.

Users Are Educated by the App About the Type of PII Processed and Security Best Practices in Using the App
During app registration, information on security best practices, such as advising user not to reveal their PIN to any third party, could be displayed. Also, during login for transaction, similar information could pop up. We observed each app for such measure during registration and transactions.

4 Findings

After analysis of acquired data from our test device, investigation revealed that none of the twelve m-banking apps saved sensitive data in the generated backup. Also, the entire apps often educated their users on security best practices. However, with the

Table 6. Summary of analysis of m-banking apps

m-banking apps	Sensitive data			App enforces device access security policies	App educates users
	Not saved in generated backup	Not held in memory	Removed when backgrounded		
Bank 1	Yes	No	No	No	Yes
Bank 2	Yes	No	No	No	Yes
Bank 3	Yes	No	No	No	Yes
Bank 4	Yes	No	No	No	Yes
Bank 5	Yes	No	No	No	Yes
Bank 6	Yes	No	No	No	Yes
Bank 7	Yes	No	No	No	Yes
Bank 8	Yes	No	No	No	Yes
Bank 9	Yes	No	No	No	Yes
Bank 10	Yes	No	No	No	Yes
Bank 11	Yes	No	No	Yes	Yes
Bank 12	Yes	Yes	No	No	Yes

Table 7. User information stored on mobile banking application after registration

Mobile applications	Username	Password	Transaction PIN	Security questions	Registered email address	Phone number	ATM card number/ type	Account number	Account name	Account type	OTP
Bank 1	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No
Bank 2	Yes	No	Yes	No	No	No	Yes	Yes	Yes	Yes	No
Bank 3	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	No
Bank 4	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Bank 5	Yes	No	No	No	No	Yes	No	Yes	Yes	No	No
Bank 6	Yes	No	No	No	No	Yes	No	Yes	Yes	No	No
Bank 7	Yes	No	No	No	No	No	Yes	Yes	Yes	Yes	No
Bank 8	Yes	No	No	No	No	No	No	Yes	No	No	No
Bank 9	Yes	No	No	Yes	No	Yes	No	Yes	No	No	No
Bank 10	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	No	No
Bank 11	Yes	No	No	No	No	No	No	No	Yes	Yes	No
Bank 12	No	No	No	No	No	No	No	No	No	No	No

Table 8. User- and application-generated data after transaction

Mobile applications	Account balance	Amount transferred	Beneficiary details	Date of transaction	Transaction time
Bank 1	Yes	Yes	Yes	Yes	Yes
Bank 2	Yes	Yes	Yes	Yes	Yes
Bank 3	Yes	Yes	Yes	Yes	Yes
Bank 4	Yes	Yes	Yes	Yes	Yes
Bank 5	Yes	Yes	Yes	Yes	Yes
Bank 6	Yes	Yes	Yes	No	No
Bank 7	Yes	Yes	Yes	Yes	Yes
Bank 8	No	Yes	No	Yes	Yes
Bank 9	Yes	No	No	No	No
Bank 10	No	Yes	Yes	Yes	Yes
Bank 11	No	No	No	No	No
Bank 12	No	No	No	No	No

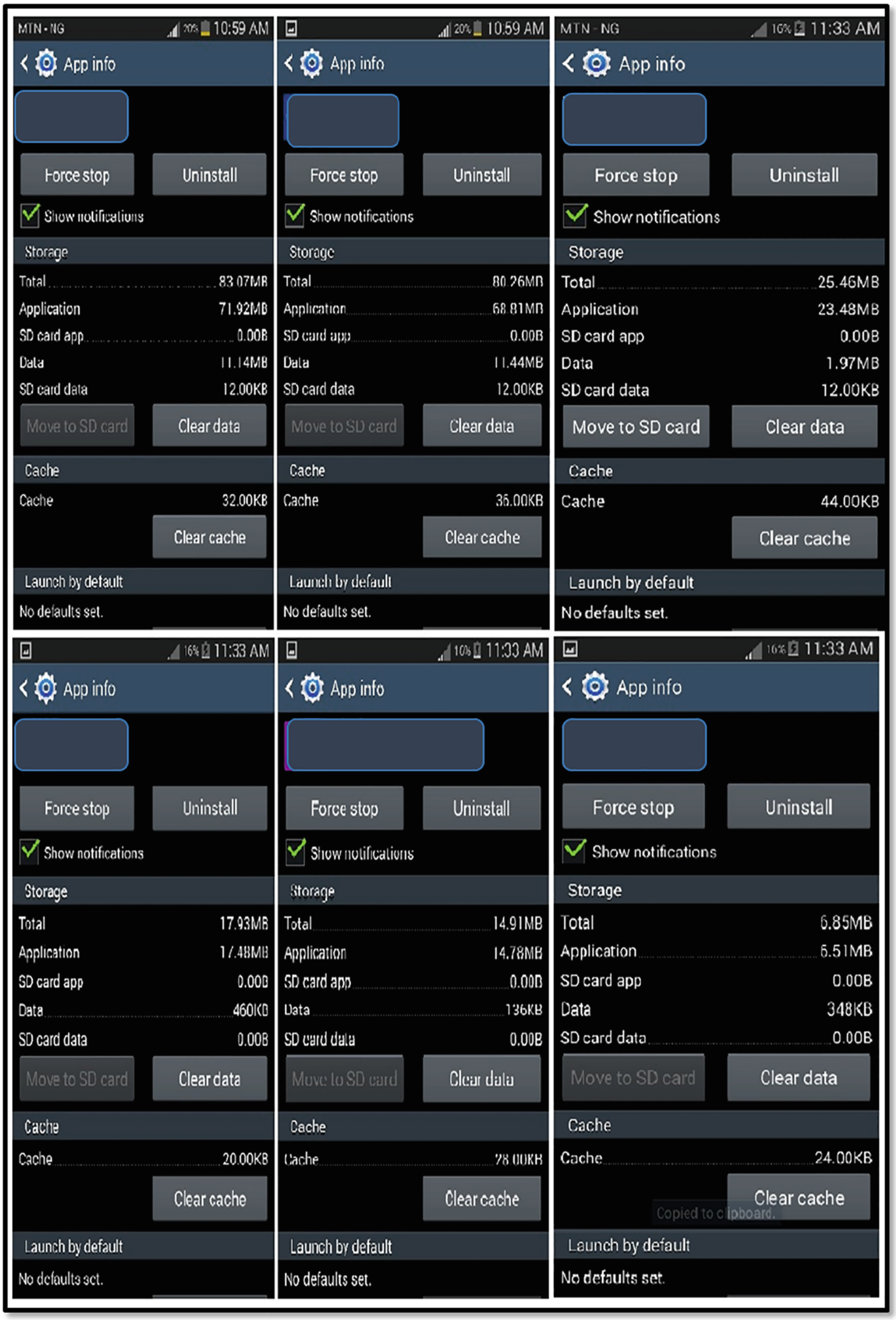

Fig. 1. Application information of six of the m-banking apps

exception of Bank 12, the apps held sensitive user data in their memory longer than necessary in the memory. Evidence of increase in the size of data in the internal memory and cache of the mobile device, after transactions, for six of the apps are presented in Fig. 1. Our findings also revealed that none of the apps removed sensitive data when backgrounded. Regrettably, it was discovered, only Bank 11 enforced any device access security policy. A summary of the findings are presented in Table 6.

Regarding sensitive data being held in the memory, data such as username, phone number, account number, and account name were displayed by most of the apps. In few of them, we were able to retrieve password, transaction PINs, security question, registered email address, ATM card number/type, account type, and OTP. Table 7 contains the user registration information retrieved from the apps.

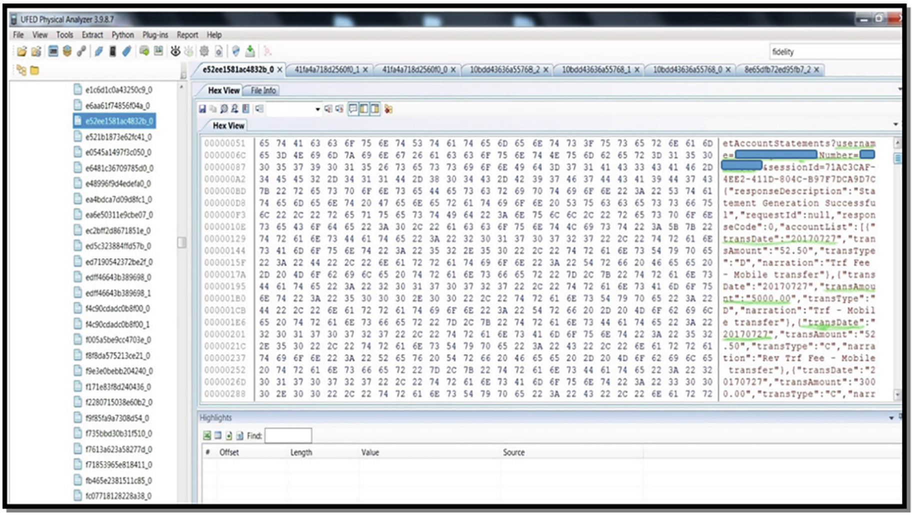

Fig. 2. Screenshot of memory dump showing user name, account number, beneficiary details, transferred amount and transaction timestamp for one of the m-banking apps

Other sensitive data generated after transaction were found. A summary of the performance of each app in this regard is presented in Table 8. We retrieved account balance, amount transferred, details of beneficiary, and date and time of transaction from Banks 1–5 and 7. Banks 6, 8–10 stored some of the data. We did not retrieve any of such data from Banks 11 and 12. Figure 2 shows the transaction-related information extracted from one of the m-banking apps.

5 Conclusion

In this study, we conducted forensic examination of twelve popular Android m-banking apps in Nigeria and assessed their performance based on five OWASP MASVS-L2 requirements. From our findings, while all of the apps performed well in two of the requirements: not saving sensitive data in backup generated by the mobile OS and educating users on security best practices, all except one of the apps held data of sensitive value, such as PII and transaction-generated data, in the memory of the test device and did not enforce any device access security policy. All the m-banking apps failed the requirement of removing sensitive data when backgrounded.

Our findings corroborate those in [20] and [31]. We also align with their recommendations on the need for app developers to consider security as a critical necessity right from the design phase and incorporate the guidelines stipulated in standard documents, such as the OWASP Mobile Security Testing Guide [17] and Mobile AppSec Verification Standard [19].

References

1. Ntantogian, C., Apostolopoulos, D., Marinakis, G., Xenakis, C.: Evaluating the privacy of Android mobile applications under forensic analysis. Comput. Secur. **42**, 66–76 (2014)
2. Statista: Number of smartphone users worldwide from 2014 to 2020 (in billions), 29 March 2019
3. Nie, J., Hu, X.: Mobile banking information security and protection methods. In: 2008 International Conference on Computer Science and Software Engineering Mobile, pp. 587–590 (2008)
4. Odumeru, J.A.: Going cashless: adoption of mobile banking in Nigeria. Arab. J. Bus. Manag. Rev. (Niger. Chapter) **1**(2), 9–17 (2013)
5. Shaikh, A.A., Karjaluoto, H.: Telematics and informatics mobile banking adoption: a literature review. Telematics Inform. **32**(1), 129–142 (2015)
6. Bankole, F.O., Bankole, O.O., Brown, I.: Mobile banking adoption in Nigeria. Electron. J. Inf. Syst. Dev. Ctries. **47**(2), 1–23 (2011)
7. Bankole, O., Cloete, E.: Mobile banking: a comparative study of South Africa and Nigeria. In: IEEE Africon 2011, Livingstone, Zambia, pp. 1–6. IEEE (2011)
8. Fenu, G., Pau, P.L.: An analysis of features and tendencies in mobile banking apps. Procedia Comput. Sci. **56**, 26–33 (2015). Elsevier Masson SAS
9. Citi: Mobile Banking One of Top Three Most Used Apps by Americans, 2018 Citi Mobile Banking Study Reveals (2018). (30 Mar 2019)
10. Juniper Research: Mobile Banking Users to Reach 2 Billion by 2020, Representing More than 1 in 3 of Global Adult Population, 30 Mar 2019
11. Elkhodr, M., Shahrestani, S., Kourouche, K.: A proposal to improve the security of mobile banking applications. In: 2012 Tenth International Conference on ICT and Knowledge Engineering A, pp. 260–265 (2012)
12. Osho, O., Yisa, V.L., Ogunleke, O.Y., Abdulhamid, S.M.: Mobile spamming in Nigeria: an empirical survey. In: 2015 International Conference on Cyberspace Governance, pp. 150–159 (2015)
13. Agwu, E.M., Carter, A.: Mobile phone banking in Nigeria: benefits, problems and prospects. Int. J. Bus. Commer. **3**(6), 50–70 (2014)
14. NCC: Monthly Subscriber Technology Data. Subscriber Statistics, 29 Mar 2019
15. Osho, O., Ajisola, T.H., Onoja, A.D., Ugwu, J.N.: Were we ready in the first place?: an analysis of cashless policy implementation in Nigeria. In: CEUR Workshop Proceedings, pp. 70–78 (2016)

16. Islam, M.S.: Systematic literature review: security challenges of mobile banking and payments system. Int. J. u- e-Serv. Sci. Technol. **7**(6), 107–116 (2014)
17. Mueller, B., Scheier, S., Willemsen, J.: Mobile Security Testing Guide (MSTG). Open Web Application Security Project (OWASP), pp. 1–412 (2019)
18. Osho, O., Ohida, S.O.: Comparative evaluation of mobile forensic tools. IJ Inf. Technol. Comput. Sci. **1**(January), 74–83 (2016)
19. Scheier, S., Willemsen, J.: OWASP Mobile Application Security Verification Standard (MASVS) version 1.1.3. Open Web Application Security Project (OWASP), 99. 1–32 (2019)
20. Chanajitt, R., Viriyasitavat, W., Choo, K.R.: Forensic analysis and security assessment of Android m-banking apps. Aust. J. Forensic Sci. **50**(1), 3–19 (2018)
21. Al Mutawa, N., Baggili, I., Marrington, A.: Forensic analysis of social networking applications on mobile devices. Digit. Invest **9**(Suppl), S24–S33 (2012)
22. Alyahya, T., Kausar, F.: Snapchat analysis to discover digital forensic artifacts on Android smartphone. Procedia Comput. Sci. **109**, 1035–1040 (2017)
23. Walnycky, D., Baggili, I., Marrington, A., Moore, J., Breitinger, F.: Network and device forensic analysis of Android social-messaging applications. Digit. Invest. **14**, S77–S84 (2015)
24. Adebayo, O.S., Sulaimon, S.A., Osho, O., Abdulhamid, S.M., Alhassan, J.K.: Forensic analysis of Kik messenger on Android devices. In: 2nd International Engineering Conference (IEC 2017), Minna, Nigeria (2017)
25. Ovens, K.M., Morison, G.: Forensic analysis of Kik messenger on iOS devices. Digit. Invest. **17**, 40–52 (2016)
26. Azfar, A., Choo, K.R., Liu, L.: An Android communication app forensic taxonomy. J. Forensic Sci. **61**(5), 1337–1350 (2016)
27. Azfar, A., Choo, K.R., Liu, L.: Forensic taxonomy of popular Android mHealth apps. In: 21st Americas Conference on Information Systems, pp. 1–19 (2015)
28. Jung, J.H., Kim, J.Y., Lee, H.C., Yi, J.H.: Repackaging attack on android banking applications and its countermeasures. Wirel. Pers. Commun. **73**, 1421–1437 (2013)
29. Bojjagani, S., Sastry, V.N.: STAMBA: security testing for Android mobile banking apps. In: Thampi, S., Bandyopadhyay, S., Krishnan, S., Li, K.C., Mosin, S., Ma, M. (eds.) Advances in Signal Processing and Intelligent Recognition Systems. AISC, vol. 425, pp. 671–683. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28658-7_57
30. Chen, S., Meng, G., Su, T., Fan, L., Xue, M., Xue, Y., et al.: AUSERA: large-scale automated security risk assessment of global mobile banking apps. arXiv:180505236, pp. 1–14 (2018)
31. Uduimoh, A.A., Ismaila, I., Osho, O., Abdulhamid, S.M.: Forensic analysis of mobile banking applications in Nigeria. i-manager’s. J. Mobile Appl. Technol. **6**(1), 9–20 (2018)
32. Srivastava, H., Tapaswi, S.: Logical acquisition and analysis of data from android mobile devices. Inf. Comput. Secur. **23**(5), 450–475 (2015)