Computer Systems Science & Engineering
DOI:10.32604/csse.2021.016787
Article

Reverse Engineering of Mobile Banking Applications

Syeda Warda Asher[1], Sadeeq Jan[1,*], George Tsaramirsis[2], Fazal Qudus Khan[3], Abdullah Khalil[1] and Muhammad Obaidullah[4]

[1]National Center for Cyber Security, Department of Computer Science & Information Technology, University of Engineering & Technology, Peshawar, 25120, Pakistan
[2]Higher Colleges of Technology, Abu Dhabi Women's College, Abu Dhabi, UAE
[3]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[4]Department of Computer Software Engineering, University of Engineering & Technology, Mardan, 23200, Pakistan
*Corresponding Author: Sadeeq Jan. Email: sadeeqjan@uetpeshawar.edu.pk
Received: 11 January 2021; Accepted: 14 February 2021

Abstract: Software reverse engineering is the process of analyzing a software system to extract the design and implementation details. Reverse engineering provides the source code of an application, the insight view of the architecture and the third-party dependencies. From a security perspective, it is mostly used for finding vulnerabilities and attacking or cracking an application. The process is carried out either by obtaining the code in plaintext or reading it through the binaries or mnemonics. Nowadays, reverse engineering is widely used for mobile applications and is considered a security risk. The Open Web Application Security Project (OWASP), a leading security research forum, has included reverse engineering in its top 10 list of mobile application vulnerabilities. Mobile applications are used in many sectors, e.g., banking, education, health. In particular, the banking applications are critical in terms of security as they are used for financial transactions. A security breach of such applications can result in huge financial losses for the customers as well as the banks. There exist various tools for reverse engineering of mobile applications, however, they have deficiencies, e.g., complex configurations, lack of detailed analysis reports. In this research work, we perform an analysis of the available tools for reverse engineering of mobile applications. Our dataset consists of the mobile banking applications of the banks providing services in Pakistan. Our results indicate that none of the existing tools can carry out the complete reverse engineering process as a standalone tool. In addition, we observe significant differences in terms of the execution time and the number of files generated by each tool for the same file.

Keywords: Reverse engineering; mobile banking applications; security analysis

1 Introduction

With the introduction of smartphones in today's ever-evolving telecom industry, mobile applications have been revolutionizing our daily lives. As per the study conducted by Lee et al. [1], there are over

1.2 billion users in the world that use smartphones for various purposes, e.g., increasing their knowledge via Internet, office work, e-commerce, online banking, social media, entertainment. With the vast use of mobile phones and the rich features they offer, many financial organizations are noticeably increasing expenditure on mobile application development to increase their productivity while delivering a more intuitive user experience. Fig. 1 shows the exponential growth trend over the past few years in android phone users in US [2].

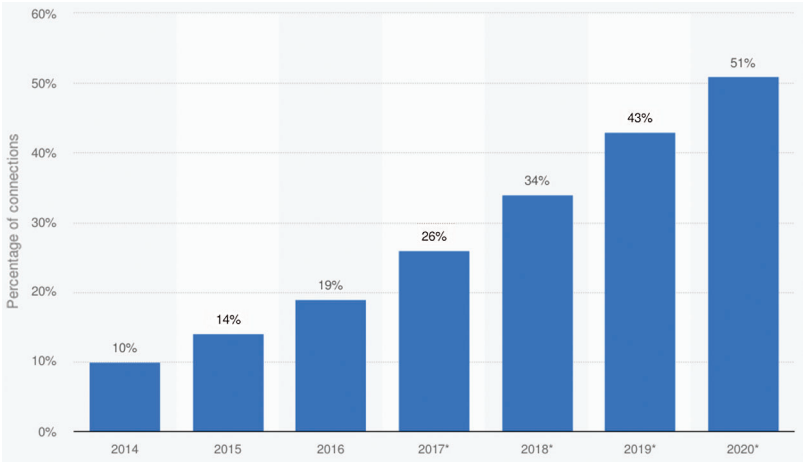

Figure 1: Growth of android users in US [2]

The rise in the use of use of mobile applications have also resulted in increase in the number of vulnerabilities in these applications. The presence of such vulnerabilities in the code eventually leads to cyber-attacks on financial applications, resulting in financial losses to banks as well as their customers. The privacy and security of these critical banking applications is of great concern as their usage is increasing rapidly. Developers must consider security requirements while developing mobile applications. Android is one of the leading and most popular mobile operating system for users as well as for businesses. For this reason, most of the developers deploy their services on android platform. Fig. 2 depicts the statistics of mobile users by their mobile connection type in Pakistan [3].

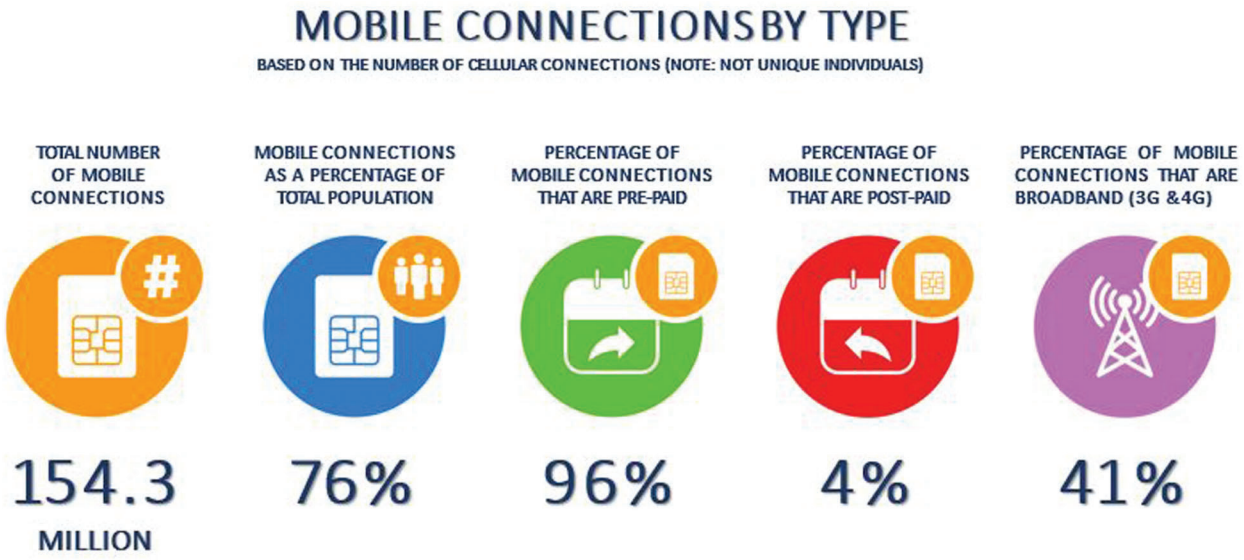

Figure 2: Mobile users based on connection type [3]

As depicted in Fig. 3, there is a significant growth in the number of android users in Pakistan [3]. Such increase has attracted more cyber-attacks and therefore the security analysis and testing of such applications is crucial. Application testing is not only recommended in the development phase, but also after deployment to ensure that the application is running error-free. Penetrations testers disguise as actual attackers and test the application for possible vulnerabilities and exploitation. One of the important stages of exploitation is reverse engineering. Reverse engineering is the process of obtaining information about the structure and implementation details of a program, which gives an insight into the working model of the application, its code structure and the associated information like dependencies on third-party applications etc. In information security, reverse engineering is used for finding vulnerabilities, creating attack vectors, and cracking applications for code analysis. This is often achieved by obtaining the code in plain text or reading it through binaries or mnemonics. It involves the extraction of design artifacts, synthesizing abstractions, and conducting further tests.

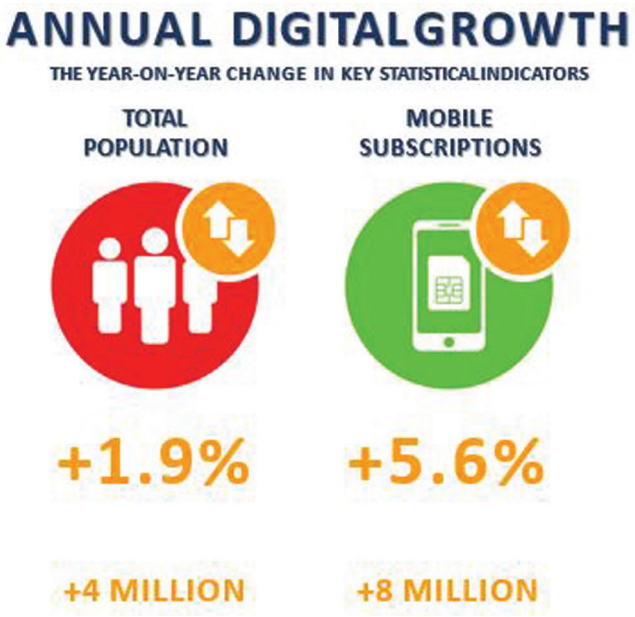

Figure 3: Growth of android users in Pakistan [3]

The focus of this research work is the in-depth analysis of reverse engineering tools for mobile applications. Mobile applications are still evolving and have limited tools for reverse engineering. Many of such tools do not provide either in-depth knowledge or the target task is achieved by using multiple tools. Furthermore, mobile being an emerging, popular and focused platform is used by most sectors to extend their services especially financial; for instance, mobile banking, telemarketing, e-commerce, social media, entertainment, and tons of others as depicted in Fig. 4. Mobile banking applications are security critical applications and any misconfiguration may cause a great impact on the user as well as the bank's business. Security features should be included when developing such banking applications. OWASP lists reverse engineering as one of the mobile applications vulnerabilities in its top 10 list [4]. The result of a case study of mobile banking applications by Zimperium [5] is shown in Fig. 5.

Our main objective of this research work is the analysis of mobile application reverse engineering tools on a data set of mobile banking applications of Pakistan. Penetration testing for web applications has been the focus of researchers in this field. However, security testing of mobile applications is a relatively new area and there is a lack of a comprehensive analysis of the tools/techniques to guide the penetration testers. The study

of reverse engineering of mobile applications for the purpose of finding security issues is important. There exist several tools for reverse engineering, however, they are not properly classified and neither evaluated by the researchers for their performance and efficiency.

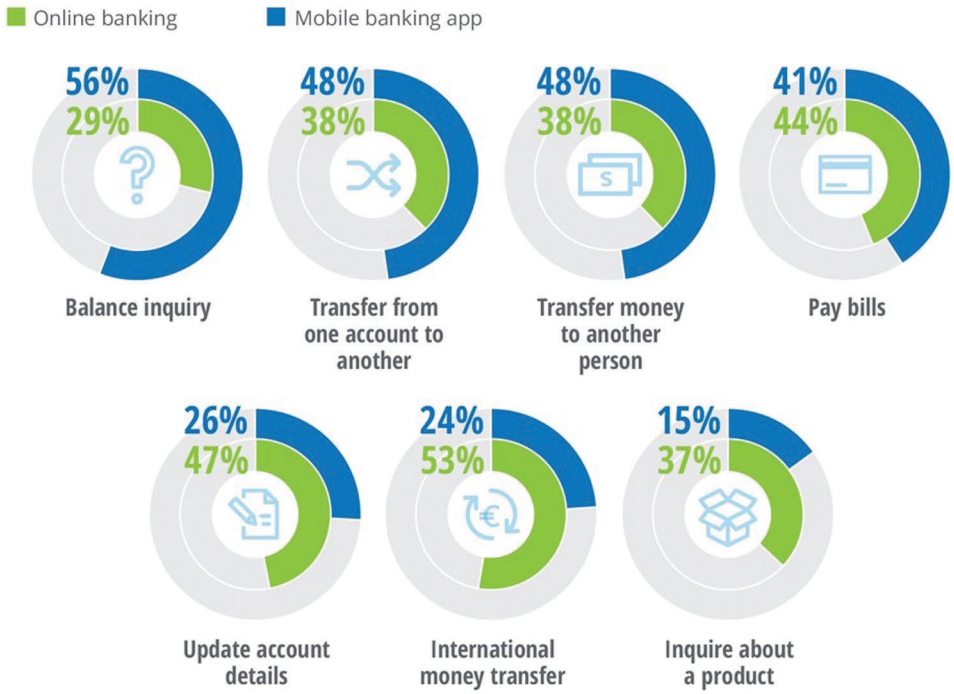

Figure 4: Use of online and mobile apps for banking activities

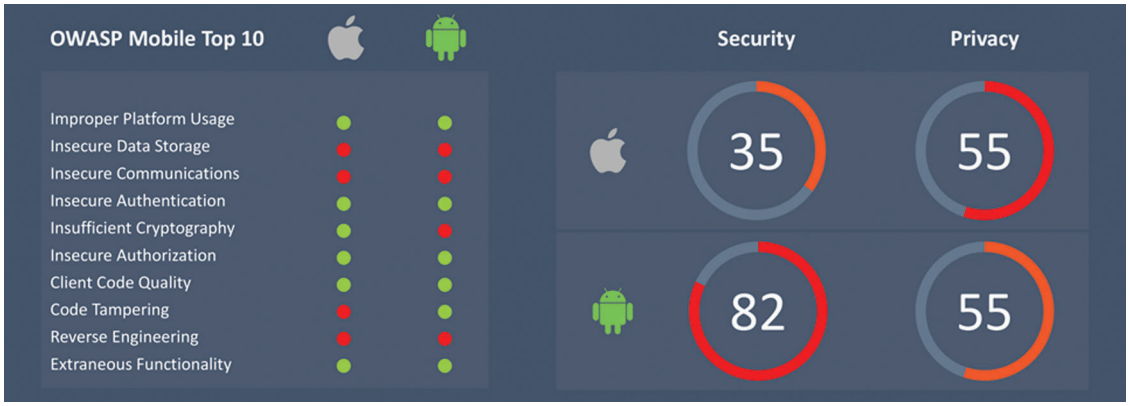

Figure 5: Security flaws in banking apps [5]

Following are the main contributions of this research work:

- A systematic collection and classification of tools used for reverse engineering of mobile applications.
- A systematic data collection of mobile banking applications (APK files) of the banks in Pakistan.
- A comprehensive study of the mobile banking applications used in Pakistan.

- A comprehensive analysis of the performance of the existing reverse engineering tools.

In this research work, we focus on answering the following research questions:

- **RQ1:** Are existing tools enough to achieve basic mobile reverse engineering tasks (from a beginner perspective)?
- **RQ2:** How user-friendly are the reverse engineering tools?
- **RQ3:** How is the performance of the tools for reverse engineering of mobile applications in terms of execution time?
- **RQ4:** How is the error-handling behavior of the tools?
- **RQ5:** Are there significant differences in the number of files generated by each tool for the same set of mobile applications?

2 Literature Review

Smartphones, being the most popular digital devices, have allowed application developers as well as various sectors to exploit the rapid growth of telecom/IT infrastructure to reach out to more clients effectively and sell their products in a hassle-free manner. Banking organizations are also utilizing this platform for better user experience as well as effective banking. Many banks have extended their services from internet banking towards mobile banking. In the same manner, users have shifted to mobile banking due to its ease of use. Figs. 6 and 7 show the increasing number of mobile banking activities by users in US [2,6]. As depicted in these figures, with each passing day, more users are getting attracted towards mobile based financial activities and the number is still increasing. The ease of use of mobile banking applications play a vital role in attracting customers to this service. In Pakistan, similar statistics are seen in the usage of mobile banking platforms among users for financial transactions. The State Bank of Pakistan's report shows that mobile banking transaction is increasing significantly as evident in Figs. 8 and 9 [7]. This number is predicted to reach 271.3 billion PKR by the end of 2019. With such growth predicted, mobile banking applications have become a popular target for security experts as well as cybercriminals.

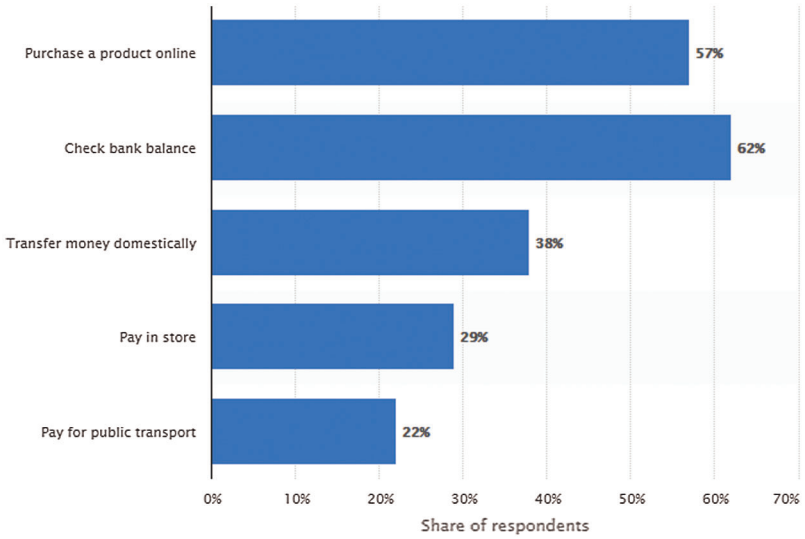

Figure 6: Usage of mobile banking services in US 2017 [2]

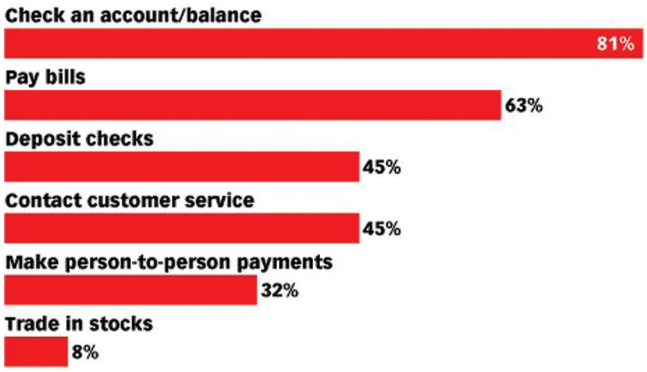

Figure 7: Mobile banking activities, US 2018 [6]

(Volume in Million & Value in Billion-PKR)

Transaction Type	Quarter-1 FY18		Quarter-2 FY18		Quarter-3 FY18		Quarter-4 FY18		Quarter-1 FY19P	
	Volume	Value	Volume	Value	Volume	Value	Volume	Value	Volume	Value
Intra-Bank Fund Transfers	1.1	33.9	1.3	41.6	1.6	51.4	1.8	59.3	1.9	59.8
Inter-Bank	0.9	34	1.1	44.2	1.4	54.8	1.6	63.4	1.6	64.8
Utilities Bill Payments	2	2.6	2.3	2.2	2.7	2.3	3.1	3.5	3.5	5.0
Misc. Payment Through Mobile	0.2	3.4	0.2	3.8	0.2	4.4	0.3	5.1	0.3	5.4
Total	**4.1**	**73.9**	**5.0**	**91.7**	**5.9**	**112.8**	**6.8**	**131.4**	**7.2**	**135.0**

Figure 8: Mobile phone banking transactions Q1FY19

(Volume in Million & Value in Billion-PKR)

Transaction Type	Quarter-3 FY18		Quarter-4 FY18		Quarter-1 FY19		Quarter-2 FY19		Quarter-3 FY19P	
	Volume	Value	Volume	Value	Volume	Value	Volume	Value	Volume	Value
Intra-Bank Fund Transfers	1.6	51.4	1.8	59.3	1.9	59.8	2.3	72.0	3.2	110.3
Inter-Bank	1.4	54.8	1.6	63.4	1.6	64.8	1.9	77.0	2.7	111.3
Utilities Bill Payments	2.7	2.3	3.1	3.5	3.5	5.0	3.9	4.3	5.3	4.7
Misc. Payment Through Mobile	0.2	4.4	0.3	5.1	0.3	5.4	0.4	6.7	0.7	45.0
Total	**5.9**	**112.8**	**6.8**	**131.4**	**7.2**	**135.0**	**8.5**	**159.9**	**11.9**	**271.3**

Figure 9: Mobile banking transaction Q3FY19

As per the State Bank of Pakistan's report [7], there exists a total of 45 banks and only 23 of them provide mobile banking services. Many of these banks find adopting mobile banking a security risk to their services. Bhatnagar et al. [8] have identified several security requirements for android mobile banking applications. Similarly, Elkhodr et al. [9] proposed measures to enhance the security of mobile banking applications by automating the authentication process.

As the usage of mobile banking applications is increasing, it is a security risk when these applications leak information about their structure, development environment, and associated information. Cybercriminals can use reverse engineering to obtain information from mobile banking applications. During reverse engineering, the coding process is reversed [10] and therefore is considered a threat from a security perspective [11]. Using reverse engineering, the tester/attacker can extract information about

the application's structure and behavior, source code, permission model, functionalities, third-party dependency and hardcoded strings. This information can further be used to upgrade, make a copy, or any other malicious purpose. There are many tools used for carrying out reverse engineering for the applications developed on various platforms [12]. Confining to reverse engineering tools for mobile-based platforms, there is a difference between the compilation and architecture of two leading platforms iOS and Android [13]. Joorabchi et al. [14] have discussed in detail the reverse engineering process regarding the iOS platform while Desnos et al. [15] have given the insight to reverse engineering for android platform. Some tools are found to be good in reversing specific format files in mobile applications. Arnatovich et al. [16] have performed a comparison of three reverse engineering tools each for a different format. The authors demonstrated the tools which perform better conversion for each format, i.e, near to the original.

Our work is similar in the context of comparing tools, however, we target the performance instead of conversion types. In addition, our focus is to use reverse engineering tools on a large set of mobile banking applications for extracting security-related information. As a result of our work, it will help others in identifying the best tool(s) for achieving their objective using minimum, feasible and efficient tools according to their requirement. As a result, security analyst can view the security lacks in mobile banking applications, e.g., what critical information can be extracted using reverse engineering process etc.

3 Methodology

The proposed methodology for reverse engineering of mobile banking applications is depicted in Fig. 10.

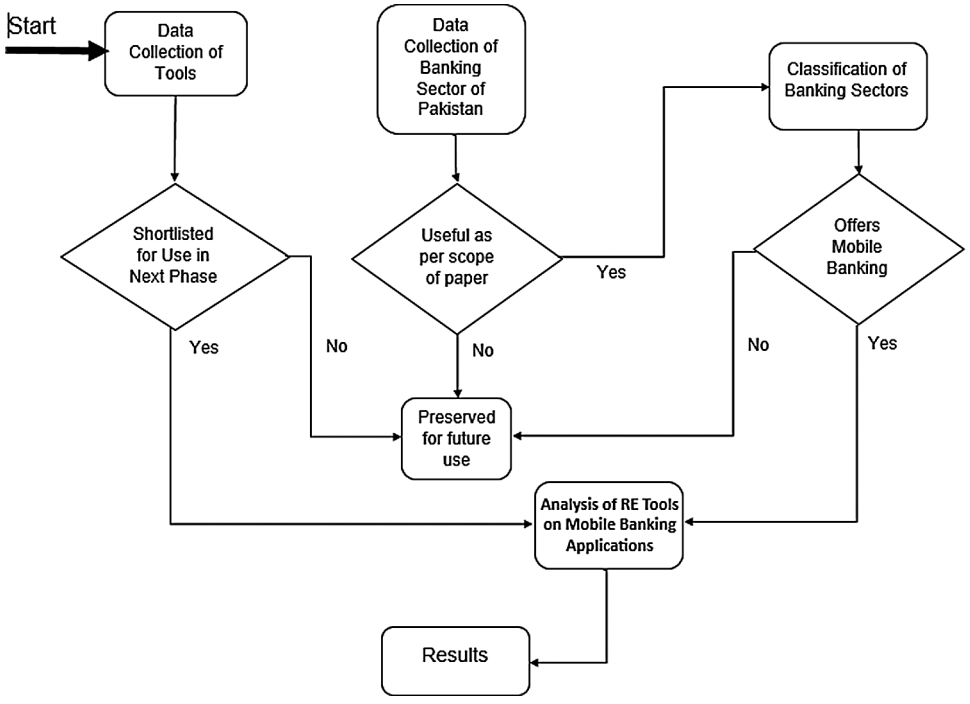

Figure 10: Proposed methodology

3.1 Data Collection

In this phase, the dataset of tools as well as banking applications was finalized. A large dataset was initially gathered from number of sources to grasp a better understanding of the subject. Eventually, the dataset was narrowed down to a limited number of mobile applications and tools to acquire quality results in a time-constrained fashion.

3.2 Data Collection of Tools

The dataset was narrowed down to a limited number of mobile applications and tools to acquire quality results in a time-constrained fashion. A dataset consisting of 42 tools and software for reverse engineering mobile applications was collected. These tools were gathered from various sources based on the literature review. We also classified these tools based on their platform for further study as we aim to study reverse engineering tools for Android platform. To limit our testbed, we selected a sample of 10 tools from our dataset. The tools were selected based on the reviews and rate of false positives. It is pertinent to mention that a mix and match of those tools were selected that offered reverse engineering of Android as well as iOS platforms. The sample of 10 out of the 42 tools is shown in Tab. 1.

Table 1: Summary of tools selected for the study

Tool	Platform	Downloaded From
Android Studio	Android	https://developer.android.com/studio
Apk Tool	Android	https://ibotpeaches.github.io/Apktool/
dex2jar	Android	https://sourceforge.net/projects/dex2jar/
Clutch	iOS	https://clutch.co/
Classdump	iOS	http://stevenygard.com/projects/class-dump/
Jdgui	Android	http://java-decompiler.github.io/
IDA	Android	http://ida.worldbank.org/
Hopper	All	https://www.hopper.com/
Andro Guard	Android	https://github.com/androguard/androguard
Smali/Basksmali	Android	https://github.com/JesusFreke/smali

3.3 Data Collection of Banking Sector

In this stage, data regarding banking sector used in Pakistan was collected. Information regarding banks that are registered with the State Bank of Pakistan was collected, assessed, and evaluated as per the scope of this research work. The main source of this data collection was the State Bank of Pakistan. A total of 45 banks were listed in this phase.

3.4 Classification of Banking Sector/Banking Applications

We analyzed the collected data of banks for the classification and further investigated the banks providing the services of Internet banking or mobile banking in Pakistan. The sole source of this information was the State Bank of Pakistan. Details of the sample analysis performed on the fragment of the data are listed in Tab. 2.

Table 2: Data collection of banking sector of Pakistan

Bank	Country of Origin
Allied Bank Limited	Pakistan
Askari Bank Limited	Pakistan
Bank Alfalah Limited	Pakistan
Bank Al-Habib Limited	Pakistan
Bank Islami Pakistan Limited	Pakistan
Burj Bank Limited	Pakistan
Citi Bank NA	US
Deutsche Bank AG	German
Dubai Islamic Bank Limited	Pakistan
Faysal Bank Limited	Pakistan
First Women Bank Limited	Pakistan
Habib Bank Limited	Pakistan
Habib Metropolitan Bank Limited	Pakistan
Industrial and Commercial Bank of China	China
Industrial Development Bank of Pakistan	Pakistan
JS Bank Limited	Pakistan
MCB Limited	Pakistan
MCB Islamic Bank Limited	Pakistan
Meezan Bank Limited	Pakistan
National Bank of Pakistan	Pakistan
NIB Bank Limited	Pakistan
SME Bank Limited	Pakistan
Samba Bank Limited	Pakistan
Sindh Bank Limited	Pakistan
Soneri Bank Limited	Pakistan
Standard Chartered Bank (Pakistan) Limited	British
Summit Bank Limited	Pakistan
The Bank of Khyber	Pakistan
The Bank of Punjab	Pakistan
The Bank of Tokyo-Mitsubishi Limited	Japan
The Punjab Provincial Co-operative Bank Limited	Pakistan
United Bank Limited	Pakistan
Zarai Taraqiyati Bank Limited	Pakistan
Advans Micro-Finance Bank Limited	Pakistan
Apna Micro-Finance Bank Limited	Pakistan

(Continued)

Table 2 (continued).

Bank	Country of Origin
Finca Micro-Finance Bank Limited	Pakistan
Khushali Bank Limited	Pakistan
Mobilink Micro-Finance Bank Limited	Pakistan
NRSP Micro-Finance Bank Limited	Pakistan
Pak-Oman Micro-Finance Bank Limited	Pakistan
Tameer Micro-Finance Bank Limited	Pakistan
The First Micro-Finance Bank Limited	Pakistan
U Micro-Finance Bank Limited	Pakistan

Based on information collected regarding mobile banking, APK files of those banks were acquired. There exists a total of 23 registered banks in Pakistan which provide mobile banking services. We were able to collect the APK files of 18 banks as depicted in Tab. 3. We executed the tools shortlisted in the previous step to assess the security of these APKs.

Table 3: Sample of mobile banking applications selected for the study

Bank	Platform	Internet/Mobile Banking
Allied Bank Limited	Both	Yes
Askari Bank Limited	Both	Yes
Bank Alfalah Limited	Both	Yes
Bank Al Habib Limited	Both	Yes
Dubai Islamic Limited	Both	Yes
Faysal Limited	Both	Yes
Habib Metro Bank	Both	Yes
JS Bank Limited	Both	Yes
MCB Limited	Both	Yes
National Bank of Pakistan	Both	Yes
Samba Bank Limited	Both	Yes
Silk Bank Limited	Both	Yes
Soneri Bank Limited	Both	Yes
SCB(Pakistan) Limited	Both	Yes
Summit Bank Limited	Both	Yes
The Bank of Punjab	Both	Yes
United Bank Limited	Both	Yes
MCB Islamic Bank Limited	Both	Yes

4 Results and Answers to Research Questions

We assessed the collected mobile applications on the reverse engineering tools for their implementation details as well as the performance of the tools. A total of 18 mobile banking applications (APKs) were assessed on the test bed of reverse engineering tools under a common environment.

Following are the main observations of our experiments:

- None of the CLI tools were able to perform the complete reverse engineering process on a mobile banking application.
- Most tools only translate from one format to another.
- External tools were required to view the generated files.
- Only one tool could handle the obfuscation, for all other tools, external de-obfuscation script was required to deobfuscate the code.
- The Apk tool generated many files, however, another tool was required for reading and extraction through directories.
- The Dex2jar tool was unable to reconvert/recompile its own translated code.
- The Jadx tool attempted to deobfuscate the code, however, it was not completely successful.
- The Enjarify tool produced better results compared to the other tools, however, it was time-consuming as it continued processing even when errors were encountered.
- Some tool despite encountering an error continued their job. The error information is logged in a text file.

For answering our research questions, we observed the following three parameters:

- Time complexity
- Error generation
- Resulting number of files

We tested the tools for their time complexity based on similar behaviors under a common environment on the banking applications testbed.

RQ1: Are existing tools enough to achieve basic mobile reverse engineering tasks (from a beginner's perspective)?

We observed that none of the tools were able to achieve the reverse engineering task as a standalone tool. Many tools just decompiled/disassembled the apk file and another tool was needed to view the source code. In addition, some tools were dependent upon other packages while others were complex to setup.

RQ2: How user-friendly are the reverse engineering tools?

The existing tools claimed to make the reverse engineering process easy, however, they did not offer a user-friendly interface for analysis of the results. Most of them simply helped to view the already decompiled/disassembled files. Similarly, the performance of these tools was slow as the user had to wait for the tool to process and show the results.

RQ3: How is the performance of the tools for reverse engineering of mobile applications in terms of execution time?

To answer this research question, we have recorded the execution times of each tool on each mobile banking application. As shown in Fig. 11, a significant difference of execution time was observed for the commands having similar nature, e.g., apk decompilation. Some of the tools took more time than others to process, however, they ended-up producing errors.

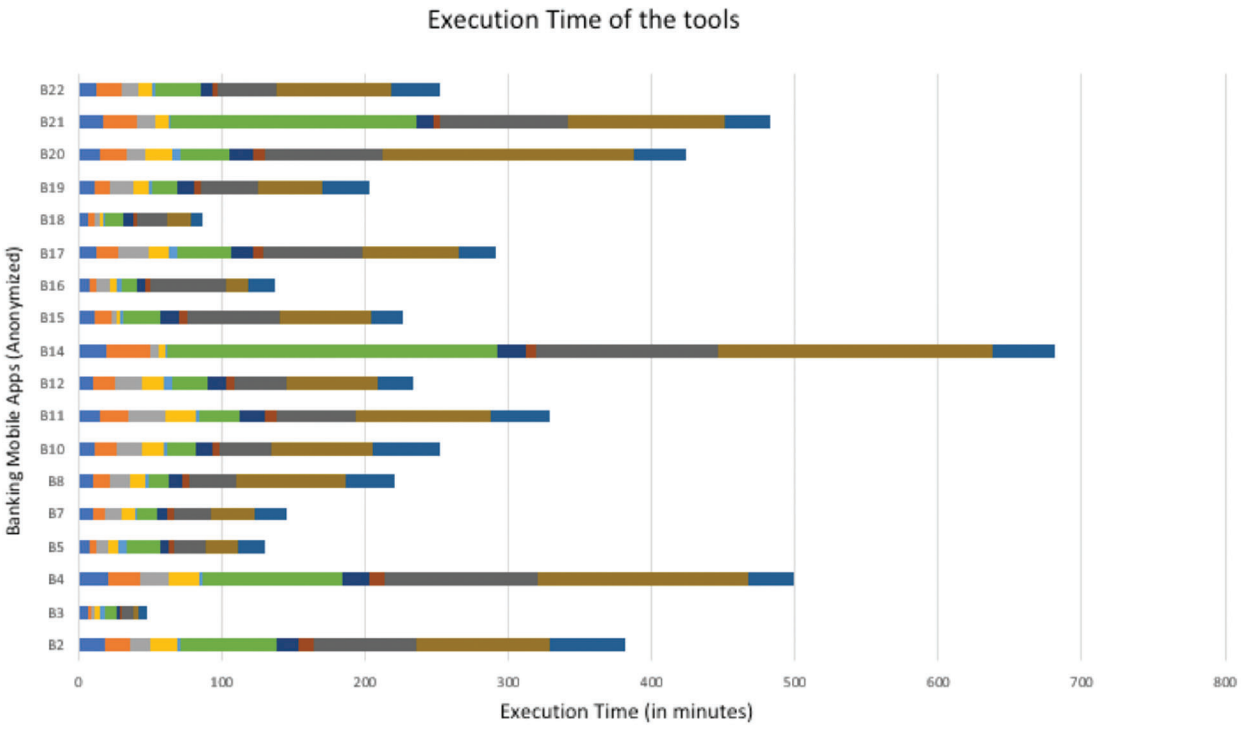

Figure 11: Execution time of tools (in minutes)

RQ4: How is the error-handling behavior of the tools?

In addition to the time complexity, we also observed whether a tool terminates or continues the reverse engineering process after encountering an error, as depicted in Fig. 12. Some of our analyzed tools were found good enough to continue performing their functions even in the presence of errors while others immediately terminated producing various error messages. Fig. 11 shows the percentage of such tools. As shown in the figure, around 22% of the tools generated some results even after errors were encountered. On the other hand, the remaining 78% of the tools aborted immediately upon encountering any type of error. Some tools logged the error information in a text file to facilitate the tester about the nature of the error.

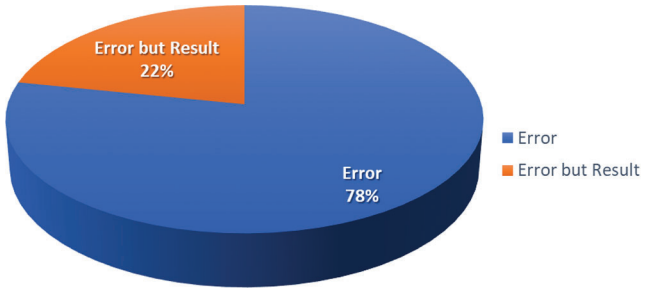

Figure 12: Results of tools (with / without errors)

RQ5: Are there significant differences in the number of files generated by each tool for the same set of mobile applications?

We observed the number of files generated by each tool for each mobile application in our dataset. It was observed that despite running similar commands, the number of files generated was different by different tools. Fig. 13 shows the number of files generated on our tested applications by each tool.

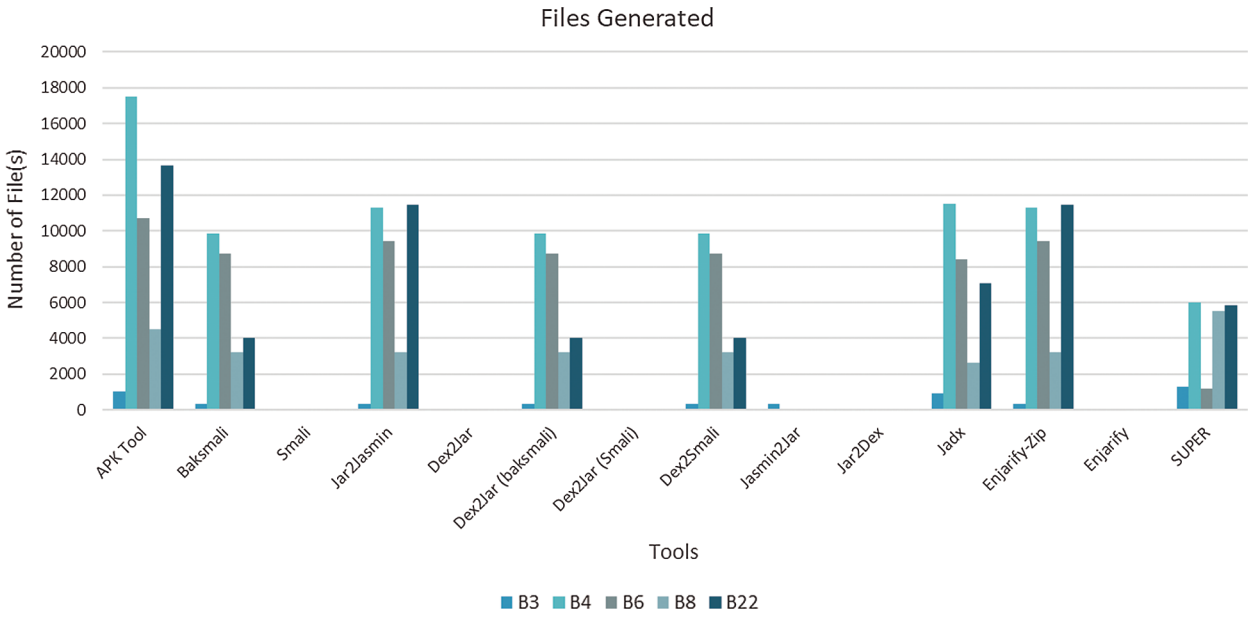

Figure 13: Number of generated files

5 Conclusion and Future Work

In this research work, we carried out an in-depth analysis of the existing well-known reverse engineering tools for the android mobile application platform in terms of their time complexity, error handling behavior, and the number of files generated. For our dataset, we selected a total of 18 mobile banking applications of the banks providing mobile banking services in Pakistan. Our results demonstrated that the tools produce different results during the reverse engineering process for the same mobile application (apk file), i.e., the number of files and directory structure produced is different. Also, there exists significant differences in the error handling methodology of the tools. Similarly, we found significant differences in the execution time of various tools for the same APK files. In future, a similar analysis of the GUI-based reverse engineering tools can be performed to evaluate their performance and help the researchers/security experts in choosing the tools for reverse engineering.

Funding Statement: The authors acknowledge the support of Security Testing-Innovative Secured Systems Lab (ISSL) established at University of Engineering & Technology, Peshawar, Pakistan under the Higher Education Commission initiative of National Center for Cyber Security (Grant No. 2(1078)/HEC/M&E/2018/707).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Lee, Y. Zhang and K. L. Chen, "An investigation of features and security in mobile banking strategy," *Journal of International Technology and Information Management*, vol. 22, no. 4, pp. 23–46, 2013.

[2] R. D. Best, "Penetration of mobile payment usage in the U.S. 2017," *Statista*, 2017. [online]. Available: https://www.statista.com/statistics/318758/mobilepayment-usage-usa/.

[3] S. Kemp, "Digital 2019 Pakistan," *Datareportal*, 2019. [Online]. Available: https://datareportal.com/reports/digital-2019-pakistan?rq=pakistan.

[4] D. Wichers and J. Williams, "OWASP Top-10 2017," *OWASP Foundation*, 2017. [Online]. Available: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/.

[5] Z. Whittaker, "Most US mobile banking apps have security and privacy flaws," *Techcrunch*, 2019. [Online]. Available: https://techcrunch.com/2019/06/11/banking-apps-security-flaws/.

[6] Morning Consult, "Consumer views on retail banking and personal finance," 2018. [Online]. Available: https://morningconsult.com/wp-content/uploads/2018/04/Morning-Consult-Consumer-Views-on-Retail-Banking-and-Personal-Finance.pdf.

[7] State Bank of Pakistan, "Payment Systems Review," 2020. [Online]. Available: https://www.sbp.org.pk/PS/PDF/PS-Review-Q3FY20.pdf.

[8] S. Bhatnagar, Y. Malik and S. Butakov, "Analysing data security requirements of android mobile banking application," in *Proc. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Cham: Springer, pp. 30–37, 2018.

[9] M. Elkhodr, S. Shahrestani and K. Kourouche, "A proposal to improve the security of mobile banking applications," in *Proc. Tenth International Conference on ICT and Knowledge Engineering*, Bangkok, Thailand, pp. 260–265, 2012.

[10] I. Wills, "Reverse engineering," in *Thomas Edison: Success and Innovation through Failure*, Cham: Springer, pp. 225–242, 2019.

[11] L. Yuntao, D. D. Soled and A. Srivastava, "Mitigating reverse engineering attacks on deep neural networks," in *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Miami, FL, USA, pp. 657–662, 2019.

[12] F. Buonamici, M. Carfagni, R. Furferi, L. Governi, A. Lapini *et al.,* "Reverse engineering modeling methods and tools: A survey," *Computer-Aided Design and Applications*, vol. 15, no. 3, pp. 443–464, 2018.

[13] R. Harrison, "Investigating the effectiveness of obfuscation against android application reverse engineering," Royal Holloway University of London, Tech. Rep. RHUL-MA-2015-7, 2015.

[14] M. E. Joorabchi and A. Mesbah, "Reverse engineering iOS mobile applications," in *Proc. 19th IEEE Working Conference on Reverse Engineering*, Kingston, ON, Canada, pp. 177–186, 2012.

[15] A. Desnos and G. Gueguen, "Android: From reversing to decompilation," in *Proc. of Black Hat*, Abu Dhabi, UAE, pp. 77–101, 2011.

[16] Y. L. Arnatovich, L. Wang, N. M. Ngo and C. Soh, "A comparison of android reverse engineering tools via program behaviors validation based on intermediate languages transformation," *IEEE Access*, vol. 6, pp. 12382–12394, 2018.