CHAPTER I

Introduction

Chapter 1

Introduction

1-1 Introduction:

A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so there are now millions of apps available. Apps are generally downloaded from application distribution platforms which are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store. Some apps are free, and others have a price, with the profit being split between the application's creator and the distribution platform.

Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

In 2009, technology columnist David Pogue stated that smartphones could be nick named "app phones" to distinguish them from earlier less-sophisticated smartphones [1] The term "app", short for "software application", has since become very popular; in 2010, it was listed as "Word of the Year" by the American Dialect Society.[2]

Mobile banking is a service provided by a bank or other financial institution that allows its customers to conduct financial transactions remotely using a mobile device such as a smartphone or tablet. Unlike the related internet banking it uses software, usually called an app, provided by the financial institution for the purpose. Mobile banking is usually available on a 24-hour basis. Some financial institutions have restrictions on which accounts may be accessed through mobile banking, as well as a limit on the amount that can be transacted. Mobile banking is dependent on the availability of an internet or data connection to the mobile device.

Transactions through mobile banking depend on the features of the mobile banking app provided and typically includes obtaining account balances and lists of latest transactions, electronic bill payments, remote check deposits, P2P payments, and funds transfers between a customer's or another's accounts[3]. Some apps also enable copies of statements to be downloaded and sometimes printed at the customer's premises. Using a mobile banking app increases ease of use, speed, flexibility and also improves security because it integrates with the user built-in mobile device security mechanisms.

From the bank's point of view, mobile banking reduces the cost of handling transactions by reducing the need for customers to visit a bank branch for non-cash withdrawal and deposit transactions. Mobile banking does not handle transactions involving cash, and a customer needs to visit an ATM or bank branch for cash withdrawals or deposits. Many apps now have a remote deposit option; using the device's camera to digitally transmit cheques to their financial institution.

Mobile banking differs from mobile payments, which involves the use of a mobile device to pay for goods or services either at the point of sale or remotely analogously to the use of a debit or credit card to effect an EFTPOS payment.

The number of banks participating in the mobile payment system reached banks and the volume of their request

For electronic money until the end of 2018 (391,742,372) pounds, and the number of subscribers

The service was about 5.6 million subscribers, an increase of 96.9% over the end of 2017 With more than 15 banking applications.

1.1.2 Mobile apps Security Testing :

Mobile Apps Security Testing is defined as a type of Software Testing that ensures software systems and applications are free from any vulnerabilities, threats, risks that may cause a big loss. Security testing of any system is about finding all possible loopholes and weaknesses of the system which might result into a loss of information, revenue, repute at the hands of the employees or outsiders of the Organization.

The goal of mobile apps security testing is to identify the threats in the system and measure its potential vulnerabilities, so the system does not stop functioning or is exploited. It also helps in detecting all possible security risks in the system and help developers in fixing these problems .

1.2 Research Problem :

The **main problem** is that there are no security standard in Sudan that govern mobile banking application **.** In case of using insecure banking applications, you can steal money from the account and transfer to another account and There are some threats: program change, data exchange and storage, violation of user privacy, In this research we check the security of the apps by testing the apps against well known security bugs, errors, backdoors.

1.3 Research Objectives:

1- Find security vulnerabilities in banking applications.
2- Identify ways to avoid exploiting the vulnerability.
3- Evaluate the security of the existing mobiles apps and produce recommendations to enhance the security of the overall systems ”service provider ”.
4- Produce recommendations to consider when designing mobile banking applications.

1.4 Scope and Limitation :

In this research,13 mobile banking application ’Android’ in Sudan will be selected for security evaluation :

Table 1:1 (show the apps and apps version of each app and the last date of the update):

Code bank	Version	Updated on
A	1.0.4	5/3/2020
B	4.3	4/7/2020
C	3.0	2/1/2020
D	2.4	14/1/2020
E	1.0.1	7/4/2019
F	1.8	1/7/2020
G	1.7.0	25/3/2020
H	2.1.4	29/6/2020
I	1.7.0	25/3/2020
J	2.0.3	6/4/2020
K	1.6.5	26/6/2020
L	2.1.4	6/2/2019
M	1.0.3	6/4/2019

1.5 Thesis Organization :

This thesis is organized as follows:

Chapter1 is the introduction to the research and it presents the problem statement and shows us the research aim and objective.

Chapter2 gives us an overview of mobile banking application

Chapter3 show the methodology “Descriptive method”

Chapter4 show the experimental results. All results discussed and analyzed in this chapter.

Finally the conclusion of research is in Chapter5 . this chapter, is the summarization of the research, recommendations, and gives some notes on the work and the future work

CHAPTER II

Literature Review and Related Work

Chapter 2

Literature review and related work

2.1 Introduction:

Electronic banking has lots of advantages for both banks and people who use online banking. Such as: it is easy, fast, cost saving and convenience to use. It has changed traditional banking industry. In addition, it is the most sensitive tasks which happened through the internet. Internet-banking, Mobile-banking, and SMS-banking are different parts of online banking .However, in this research Mobile banking has been studied.

2.2 Mobile banking:

Mobile banking and Internet banking are very similar,The applications of many smartphones connect you directly to your bank, allow you to transfer money, and some banks even allow you to make deposits by taking a picture of the check [4].

Mobile devices, smartphones, and tablets can be taken here and there easily. Furthermore, they provide users access to personal and financial data easy via applications that allow the movement also locally storage of data on the devices and allow data to be sent and to be stored with a third parts.

Through the web browser on the mobile phone, to achieving the bank's web page via text messaging, or by using an application downloaded to the mobile phone, the mobile banking can be done [5].

Bank management technologies are among the major changes in internal banking systems that also have exercised a positive influence on banking achievement and propriety[6].

This research provide an overview of mobile banking security challenges, and a key for reading and interpreting them. In addition, this research presents a number of interesting findings, including mobile banking challenges of payment security and application of mobile banking.

2.3 Benefits of mobile banking:

With mobile banking, customers can do their banking activities anytime, anywhere, cheaper and other[7]. the benefits of mobile banking:

- Saving time and saving energy
- Easy to use
- Reduce cost
- More suitable than internet-banking

2.4 vulnerabilities of mobile banking

One of the new challenges of mobile banking is the online threats, mobile user always active on online downloading various applications, song and official mail or other personal files over internet ,example vulnerabilities :

2.4.1) Distributed Denial of service (DDOS) Attack

DDOS attack is ranked as third highest threat as FBI said. DDOS is the most common attack of banking system. DDOS attack orbit the attack to target system. Before an attack is happen, attacker will attack network by scanning open ports [8]

2.4.2) Malware

Malware is the term for maliciously crafted software code. Moreover, it is possible to perform the following operations for this type of malicious software Account information theft [9]:

- Fake web site substitution
- Account hijacking

2.4.3) TCP/IP Spoofing

Here, an attacker gains unauthorized access to a mobile device or a network by making a malicious message has come from a trusted machine by "spoofing" the IP address of that machine [10].

2.4.4) Backdoors

Access to a mobile program that avoided security mechanisms is backdoor. A programmer may sometimes install a back door so that the program can be accessed for troubleshooting or other purposes. Anyhow, back doors have been used by attackers to install themselves, as chunk of an exploit [11].

2.4.5) Tampering

Tampering is an intentional modification of message in a way that would make them harmful to the bank [12].

2.4.6) Exploits

Exploit is a piece of software, or a data which acts as a bug or vulnerability in order to matter surprising behavior to exist on computer software, or hardware [13].

2.4.7)Social Engineering and Trojans

Trojans act as no authorized programs. Can delete, block, modify, and copy data. However, Trojan is not like a viruses and worms, it is not able to self-replicate [14]

2.5 literature review and previous studies :

These are some studies that have dealt with the issue of security in banking application :

2.5.1 Proposed Framework for Securing Mobile Banking Applications from Man in the Middle Attacks:

This dissertation proposes a defense-in-depth strategy for implementing controls against man-in-the middle attacks on mobile phone based banking transactions [15].

The proposed The framework can be implemented at each level:

I. Client level
II. Communication level
III. Server level

2.5.2 Security of Mobile Banking:

This paper discusses some of these security shortfalls, such as security problems with GSM network, SMS/GPRS protocols and security problems with current mobile banking solutions. This paper discusses the SMS and GPRS proposed solutions for these problems. The results from these proposed solutions have proven to provide secure and economic communications between the mobile application and the bank servers .The proposed solutions allow the users to bank communication using secure SMS and GPRS [16].

2.5.3 Security Challenge and Issue of Mobile Banking in Republic of Uzbekistan: A State of Art Survey:

Overview of the current state of online banking in Uzbekistan. This work is to classify and analyze the Security issues and challenges in Mobile banking in Uzbekistan. Majority of the customers in Uzbekistan are using online banking or ATM. However, around 40% of the customers are using the mobile banking where the remaining people 60% do not use this technology[17].

2.5.4 Securing Online Banking Services against Man in the Middle Attacks by use of two Factor Authentication.

this paper trying to enhance security of online banking against man in middle attacks using two factor authentication of first password encrypted combined with **one time password**. The two factor authentication was chosen in this paper because this type of authentication is considered faster, quicker and cheaper to set up and maintain[18].

2.5.5 Mobile Authentication Secure Against Man-In-The-Middle Attacks

This paper propose Mobile-ID, a protocol which prevents MITM attacks while keeping the human outside the security loop The proposed protocol carries the context information of the man in the middle from the mobile client to the Mobile-ID server which then compares this information with

the information belonging to the intended service provider and stops the protocol by notifying the mismatch. currently working on implementing the **MobileID protocol** and integrating it with the OpenID standard[19].

2.5.6 The Wild West Of Mobile Security

The intent of this research was to measure where the mobile banking applications stand compared to the mobile payment applications (used as a reference point guiding to better illustrate the comparison in security). **Only 5% of the applications analysed came close to this standard**[20].

2.5.7 Vulnerabilities in Banking Transactions with Mobile Devices Android:

mobile vulnerabilities are mitigated in different versions of the Android operating system, the attacker is at the forefront to take advantage of the minimum flaw that is in the current versions and thus act deliberately and especially violate banking applications.

The systematic review allowed to know the frequent vulnerabilities in the banking transactions by Android mobile devices, usually the user is not aware of what they happen, the ones that stand out are :

Banking Phishing, Trojans and injections, unsafe storage in the cloud, campaigns to violate the ANDROID platform, insufficient protection when circulating data, insecurity in servers, lack of protection in the transport

With respect to the incidence of use of banking transactions with smartphones in users, the increase in technology has partly facilitated their way of life due to the great benefits that smartphones have generated in their daily performance, the change in routine of users who use SMARTPHONES, has had a positive and largely negative impact, simply because they do not know deeply the good use of these smartphones and therefore are the victims of many bank frauds and information leakage [21].

2.6 Summary of Previous Studies:

Previous studies presented the threats facing mobile banking applications and suggested some ways to reduce or eliminate these threats by suggesting some methods and systems that can be used in the mobile banking application.

Based on these studies, a study can be made that summarizes mobile banking applications in Sudan by presenting the threats facing the applications and how to reduce these risks.

CHAPTER III

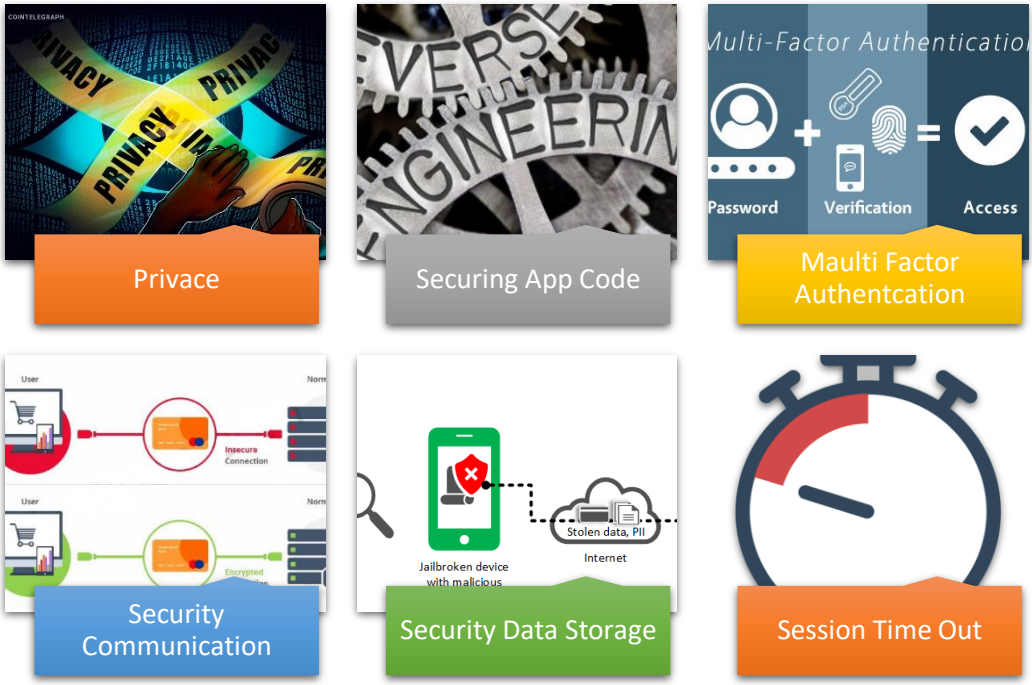
Methodology

Chapter3

Methodology

3.1 introduction :

This chapter explains methodology "descriptive method" which will be followed in this research to implement and evaluate our approaches , the following methodology steps will be used as presented figure 3:1 (evaluation factor):



3.2 privacy :

Meaning of **privacy** in English "someone's right to keep their personal matters and relationships secret "[22]

3.2.1 Important of privacy:

Privacy is the ability to control who can access information about our private life and our activities.

Privacy is important because:

- Privacy gives us the power to choose our thoughts and feelings and who we share them with.
- Privacy protects our information which we do not want it to be shared publicly (such as health or personal finances).
- Privacy helps protect our physical safety (if our real time location data is private).

- Privacy helps protect us as individuals, and our businesses, against entities we depend on or that are more powerful than us.
- Privacy is tied to Freedom…. Could we really be free – and have free will – without Privacy? [23]

3.2.2 Privacy In android:

The purpose of a permission is to protect the **privacy** of an Android user. Android apps must request permission to access sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet). Depending on the feature, the system might grant the permission automatically or might prompt the user to approve the request [24].

If the device is running Android 6.0 (API level 23) or higher, and the app's targetSdkVersion is 23 or higher, the user isn't notified of any app permissions at install time. The app must ask the user to grant the dangerous permissions at runtime. When the app requests permission, the user sees a system dialog (as shown in figure 3:2, left) telling the user which permission group the app is trying to access. The dialog includes a Deny and Allow button .

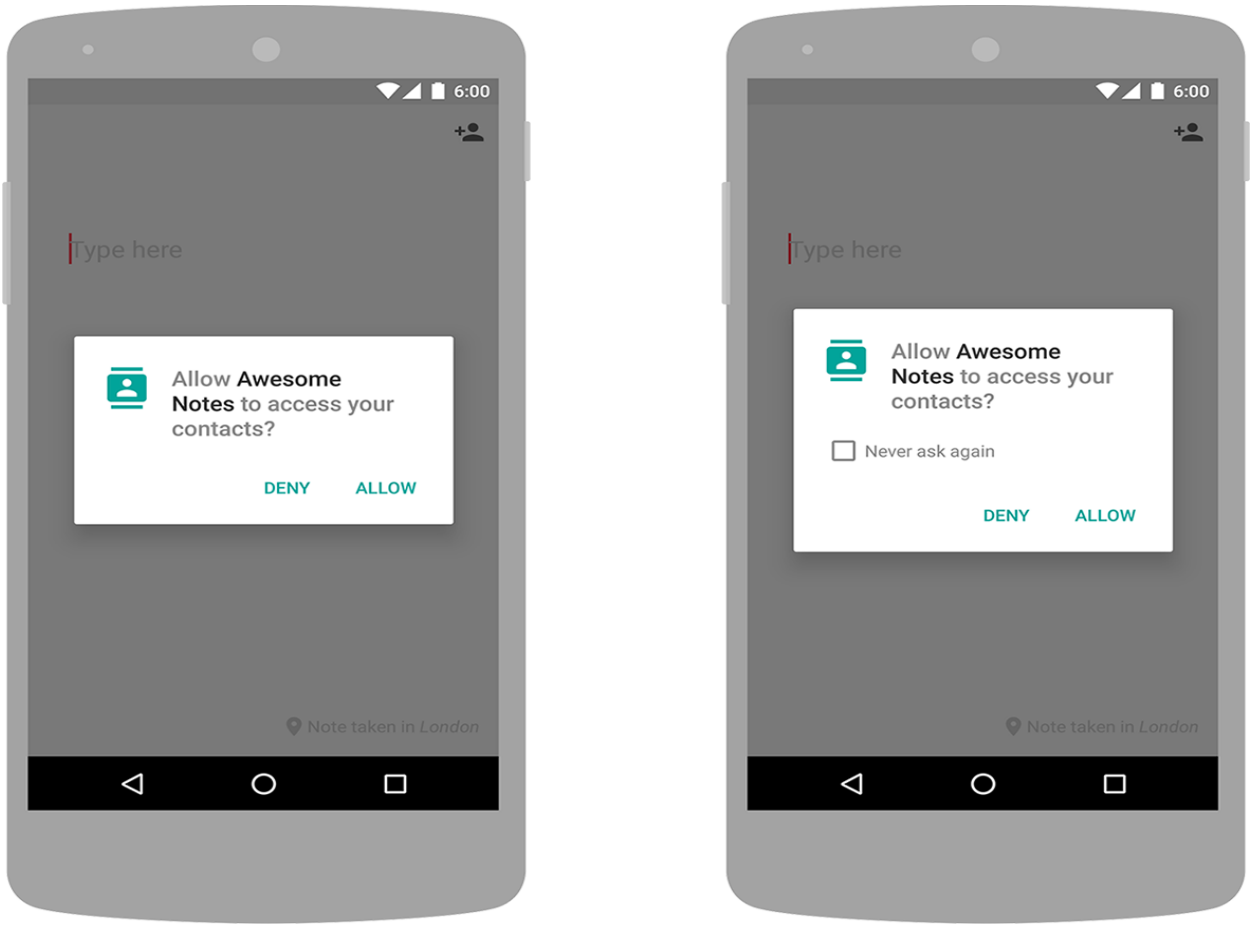


Figure 3:2(android permission) Initial permission dialog (left) and secondary permission request with option to turn off further requests (right).

If the user checks the Never ask again box and taps Deny, the system no longer prompts the user if you later attempt to requests the same permission.

Even if the user grants your app the permission it requested you cannot always rely on having it. Users also have the option to enable and disable permissions one-by-one in system settings.

3.2.2.1 One-time permissions:

Starting in Android 11 (API level 30), whenever your app requests a permission related to location, microphone, or camera, the user-facing permissions dialog contains an option called Only this time. If the user selects this option in the dialog, your app is granted a temporary one-time permission.

3.2.2.2 Protection levels:

Permissions are divided into several protection levels. The protection level affects whether runtime permission requests are required.

There are three protection levels that affect third-party apps: normal, signature, and dangerous permissions.

- **Normal permissions :**

  Normal permissions cover areas where your app needs to access data or resources outside the app's sandbox, but where there's very little risk to the user's privacy or the operation of other apps. For example, permission to set the time zone is a normal permission.

  If an app declares in its manifest that it needs a normal permission, the system automatically grants the app that permission at install time. The system doesn't prompt the user to grant normal permissions, and users cannot revoke these permissions.

- **Signature permissions :**

  The system grants these app permissions at install time, but only when the app that attempts to use a permission is signed by the same certificate as the app that defines the permission example share data between two app .

- **Dangerous permissions :**

  Dangerous permissions cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps. For example, the ability to read the user's contacts is a dangerous permission. If an app declares that it needs a dangerous permission, the user has to explicitly grant the permission to the app. Until the user approves the permission, the app cannot provide functionality that depends on that permission.

3.2.3 Privacy In iOS:

Requesting Permission

Users must grant permission for an app to access personal information, including the current location, calendar, contact information, reminders, and photos. Although people appreciate the convenience of using an app that has access to this information, they also expect to have control over their private data. For example, people like being able to automatically tag photos with their physical location or find nearby friends, but they also want the option to disable such features [25].

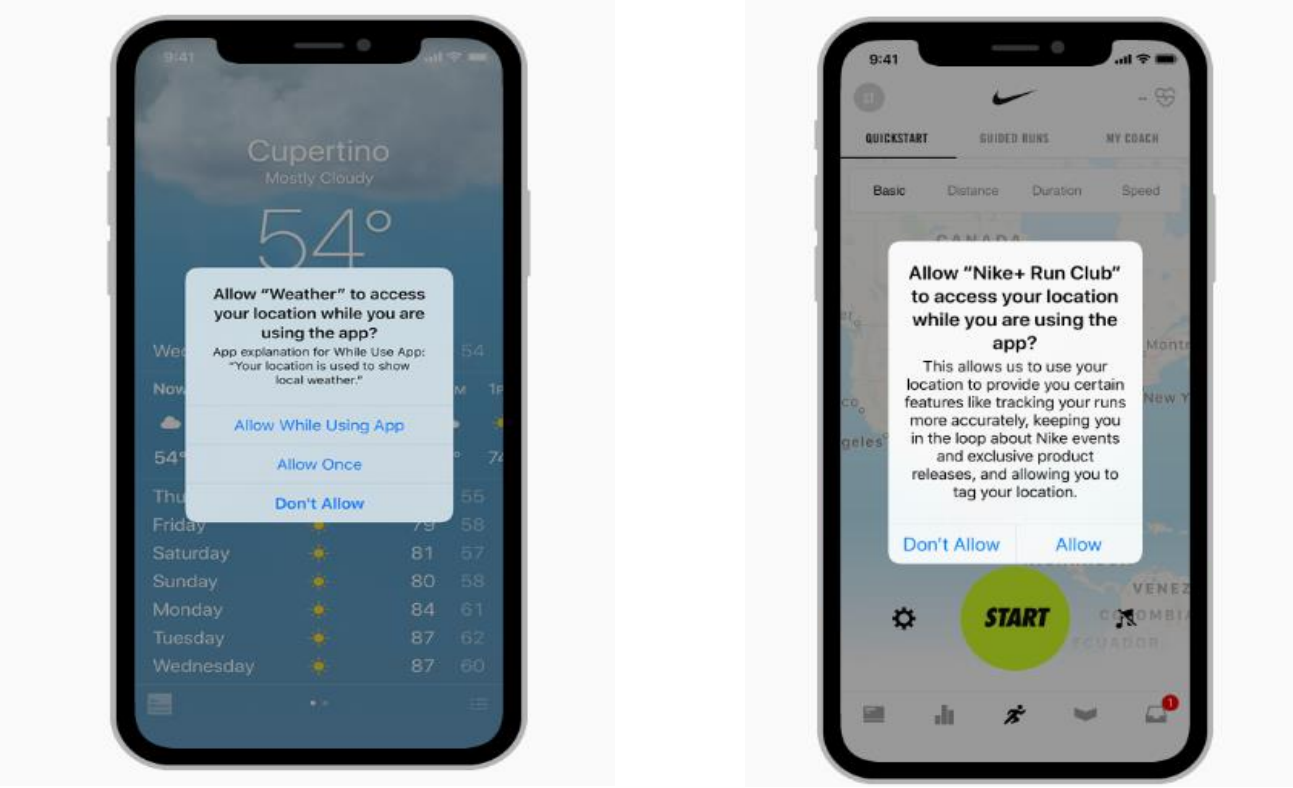


Figure 3:3(IOS permission)

3.3 securing app code (Reverse Engineering):

"Reverse engineering, also called **Black engineering**, is the process of extracting knowledge or design information from anything man made and reproducing it or reproducing anything based on the extracted information. The process often involves disassembling something (a mechanical device, electronic component, computer program, or biological, chemical, or organic matter) and analyzing its components and workings in detail" So basically, **Reverse Engineering an Android application** aid us in discovering and understanding the complete working of the application by

learning its operation, structure, and functions. Reverse engineering of Android applications will help us to:

- Read the code.
- Understand the code.
- Find loopholes in the code level.
- Locate sensitive data which might be hardcoded into the application's code.
- Migration of applications during a change to a new hardware platform
- Perform Malware Analysis.
- Modify the existing code/functionality of an existing application.

There are two processes involved in reverse engineering: Disassembly and Decompilation

Disassembly is the process of translating machine language into assembly language. The output of a disassembler is often formatted for human readability rather than focusing on suitability for input to an assembler.

Generally speaking, Decompilation is the inverse of compilation. Here we are translating an executable file into a source code with a more readable format, i.e. higher level language. It's not possible to have a fully automated Decompilation. No decompiler can get the exact source code that the developer wrote.

There are numerous reverse engineering tools:

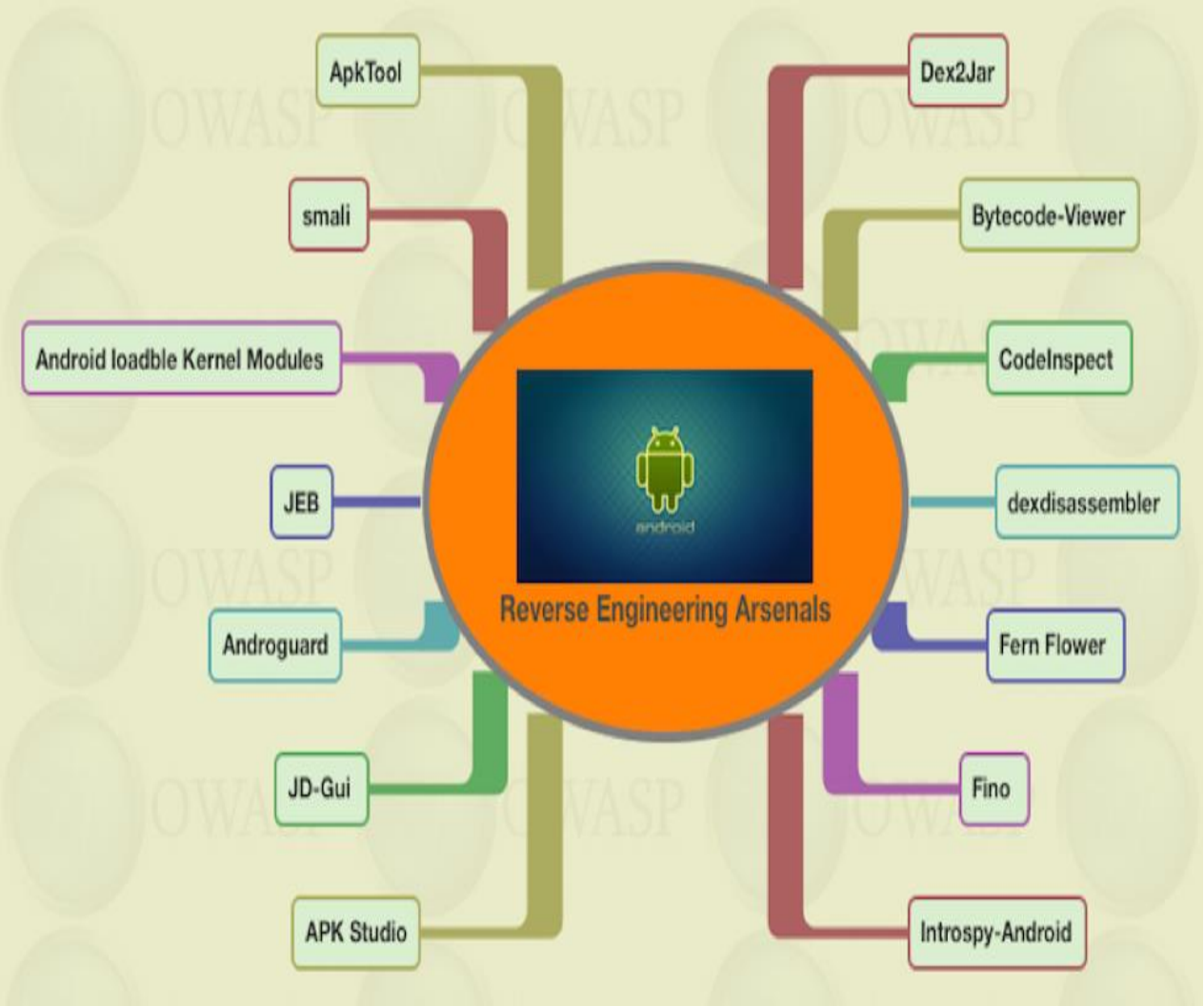


Figure 3:4(reverse engineering tools)

We will cover some of the common ones which we be used , like apktool, Dex2jar, JD-Gui etc.

3.3.1 APK format :

The **APK, *Android Application Package,*** a bundle is the format used to package any application you develop or that you can get from Google Play Store or any other channel. In other words, for each application present on your device, there is a corresponding APK file (this is true also for pre-installed applications).

An APK file is essentially a ZIP file, so you can get one, rename it and then extract it to have access to its content.

Table 2:1 (content of android application)[24]

Entry	Notes
AndroidManifest.xml	The **binary xml** file that provides information that a device needs in order to run the app.
classes.dex	The application code compiled in the dex format.
resources.arsc	Binary XML file containing precompiled application resources.
res/	Folder containing resources that are not compiled into resources.arsc file
assets/	This folder contain applications' raw assets. AssetManager Provides access to these asset files
META-INF/	It contains the MANIFEST.MF file, which stores meta data about the contents of the JAR. The signature of the APK is also stored in this folder.
lib/	This folder contains compiled the code – i.e. native code libraries.

3.3.2 NIST Special Publication 800-163 Vetting the Security of Mobile Applications

If the App is stored on an unsecured machine, unauthorized users could access the App, leading to potential licensing violations (e.g., if an unauthorized user copies a licensed App for their personal use) and integrity issues (e.g., if the app is modified or replaced with another App by an attacker). In addition, violations of intellectual property may also arise if unauthorized users access the source code of the App

3.4 Multi Factor Authentication:

the traditional, not so secure way to log in to your bank account: enter your username and that familiar password you probably use for most of your online accounts. Then, you're in. You can go about your business.

Not so fast! If you're one of the 54% of consumers who, according to TeleSign, [26]use five or fewer passwords for all of their accounts, you could create a "domino effect" that allows hackers to take down multiple accounts just by cracking one password. The good news? There's an easy way to better protect your accounts (which contain a lot of personal information) with **multi-factor Authentication (MFA)** [27].

Multi-factor authentication :

is an authentication method in which a computer user is granted access only after successfully presenting two or more pieces of evidence (or factors) to an authentication mechanism: knowledge (something the user and only the user knows), possession (something the user and only the user has), and inherence (something the user and only the user is).

A good example of two-factor authentication is the withdrawing of money from an ATM; only the correct combination of a bank card (something the user possesses) and a PIN (something the user knows) allows the transaction to be carried out.

Figure 3:5(multi factor authentication)

3.5 security communication:

3.5.1 Communications security is the system of preventing unauthorized interceptors from accessing telecommunications [28] in an intelligible form, while still delivering content to the intended recipients.

What is HTTP?

Full form of HTTP is Hypertext Transfer Protocol. HTTP offers set of rules and standards which govern how any information can be transmitted on the World Wide Web. HTTP provides standard rules for web browsers & servers to communicate.

What is HTTPS?

HTTPS stands for Hyper Text Transfer Protocol Secure. It is highly advanced and secure version of HTTP. It uses the port no. 443 for Data Communication. It allows the secure transactions by encrypting the entire communication with SSL. It is a combination of SSL/TLS protocol and HTTP. It provides encrypted and secure identification of a network server and used EC Diffie-Hellman Algorithm Is a key agreement protocol that allow two parties , each having an elliptic-curve public- private key pair, to establish a shared secret over an insecure channel .

.

Discuss the importance of communication :

- Iso / ts 12812-5.
- NIST - Vetting the Security of Mobile Applications.
- SANS Institute - Information Security Reading Room.
- Visa card .
- Master card.

3.6 securing data storage :

3.6.1 Storage Encryption :

involves encrypting data while it passes to storage devices, such as individual hard disks, tape drives, or the libraries and arrays that contain them. Using storage level encryption along with database and file encryption goes a long way toward offsetting the risk of losing your data. Like network encryption, storage encryption is a relatively blunt instrument, typically protecting all the data on each tape or disk regardless of the type or sensitivity of the data [29].

Although using storage encryption is a good way to ensure your data is safe by default in case it is lost, adopting a more granular approach and encrypting at the level of individual files, volumes, or columns in a database may be necessary, particularly if data is shared with other users or is subject to specific audit requirements.

3.6.2 Data storage security principles:

At the highest level, data storage security seeks to ensure "CIA" – confidentiality, integrity, and availability.

- **Confidentiality:** Keeping data confidential by ensuring that it cannot be accessed either over a network or locally by unauthorized people is a key storage security principle for preventing data breaches.

- **Integrity:** Data integrity in the context of data storage security means ensuring that the data cannot be tampered with or changed.
- **Availability:** In the context of data storage security, availability means minimizing the risk that storage resources are destroyed or made inaccessible either deliberately – say during a DDoS attack – or accidentally, due to a natural disaster, power failure, or mechanical breakdown[30].

3.6.3 protect data storage assets

The relevant international standard for storage security is ISO/IEC 27040, which calls for the application of physical, technical and administrative controls to protect storage systems and infrastructure as well as the data stored within them. It notes that these controls may be: preventive; detective; corrective; deterrent; recovery; or compensatory in nature.

The bottom line, according to the Storage Networking Industry Association (SNIA) is that ISO/IEC 27040 defines best practices that ultimately set the minimum expectations for storage security.

3.6.4 Data storage security: Physical controls

Physical controls are designed to protect storage resources and the data they contain from physical, as opposed to logical, access by unauthorized or malicious persons.

These physical controls come in many forms but may include:

- Guards or other security personnel monitoring data centers and storage resources to prevent unauthorized access
- CCTV monitoring with video retention
- Access controls such as biometric readers or smart card readers to prevent unauthorized access, along with anti-tailgating/anti pass-back turnstile gates that permit only one person to pass through after authentication
- Internal environment monitoring using systems such as temperature sensors and smoke detectors
- Alternative power sources such as a backup generator

3.6.5 Data storage security: Technical controls

Technical controls include many of the security procedures that are familiar to IT security professionals such as network perimeter security measures, intrusion detection and prevention systems, firewalls, and anti-malware filtering.

SNIA recommends considering:

- Incorporating storage considerations into policies after identifying the most sensitive and business-critical data categories and their protection requirements
- Integrating storage-specific policies with other policies where possible
- Addressing data retention and protection
- Addressing data destruction and media sanitization
- Ensuring that all elements of storage infrastructure comply with policies

3.7 Session Time Out:

3.7.1 Description of the session timeout:

Session timeout represents the event occurring when a user do not perform any action on a web site during interval (defined by web server). The event, on server side, change the status of the user session to 'invalid' (ie. "not used anymore") and instruct the web server to destroy it (deleting all data contained into it) **[31]**.

impact of the session timeout on security and best practices

Session timeout define action window time for a user thus this window represents, in the same time, the delay in which an attacker can try to steal and use a existing user session…

For this, the best practices to :

- Set session timeout to the minimal value possible depending on the context of the application.
- Avoid "infinite" session timeout.
- Prefer declarative definition of the session timeout in order to apply global timeout for all application sessions.
- Trace session creation/destroy in order to analyse creation trend and try to detect anormal session number creation (application profiling phase in a attack).

3.7.2 The reason the session timeouts are limited to 15 minutes is twofold:

1. Sessions require server memory to remain active. We try to keep a lid on the amount of memory sessions consume, for obvious reasons. Ideally, a spike in traffic (and sessions) should not cause the server to run out of memory suddenly, so we try to keep a fair amount ready - which means the average number of sessions needs to be managed closely[32].
2. Maintenance on the servers means that they'll have to be restarted or shutdown. Any active sessions at that time would be disrupted. To avoid that, we normally use a polite shutdown policy, where the server no longer accepts new sessions and only serves old ones. At some point, they will all expire naturally, and site visitors notice nothing is wrong. This process requires a lot of time, as it is. Increasing the session time out may [32].

CHAPTER IV

Experimental Result and Discussion

Chapter 4

Experimental result and discussion

4.1 Introduction

In this chapter, a security assessment will be made on the application of banking mobility in terms of privacy, reverse engineering, verification factors, connection security, securing stored data and session time all these factor pass by two step " Testing – evaluation "

4.2 privacy :

Every app project must have an Android Manifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

Among many other things, the manifest file is required to declare the following[24]:

- The app's package name, which usually matches your code's namespace. The Android build tools use this to determine the location of code entities when building your project. When packaging the app, the build tools replace this value with the application ID from the Gradle build files, which is used as the unique app identifier on the system and on Google Play.

- The components of the app, which include all activities, services, broadcast receivers, and content providers. Each component must define basic properties such as the name of its Kotlin or Java class. It can also declare capabilities such as which device configurations it can handle, and intent filters that describe how the component can be started.

- The permissions that the app needs in order to access protected parts of the system or other apps. It also declares any permissions that other apps must have if they want to access content from this app.

- The hardware and software features the app requires, which affects which devices can install the app from Google Play.

4.2.1 Step one testing :

1- Bank (A)

number of permissions =19

This application requires critical permission example :

1. calendar(read,modify,send email without owner's knowledge)
2. camera (take pictures and videos)
3. Bluetooth (access setting, pair)

4. use biometric hardware (fingerprint)
5. location (GPS, access approximate location).
6. Storage (modify, delete , read).
7. Telephone (read phone status and identity)

2- Bank (B)

number of permissions =17

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contact (read ,find)
3. Location (GPS, access approximate location)
4. Storage (modify, delete , read).
5. Bluetooth (access setting, pair)
6. Flashlight .

3- Bank (C)

number of permissions =7

This application requires critical permission example :

1. camera (take pictures and videos)
2. location (GPS, access approximate location).
3. Storage (modify, delete , read).

4- Bank (D)

number of permissions =11

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read)
3. Location (GPS, access approximate location
4. Telephone (read phone status and identity)

5- Bank (E) number of permissions =11

This application requires critical permission example :

1. camera (take pictures and videos)
2. location (GPS, access approximate location).
3. Telephone (read phone status and identity).

4. Storage (modify, delete , read).

6- Bank (F)

This application requires critical permission example :

number of permissions = 9

1. camera (take pictures and videos).
2. Location (GPS, access approximate location
3. Telephone (read phone status and identity, directly call)
4. Storage (modify, delete , read).

7- Bank (G)

number of permissions =5

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read)

8- Bank (H)

number of permissions =7

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read)
3. Storage (modify, delete , read).

9- Bank (I)

number of permissions =5

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read)

10- Bank (j)

number of permissions =2

This application not requires critical permission

11- Bank (K) number of permissions =15

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read)
3. Location (GPS, access approximate location)
4. Storage (modify, delete , read).
5. Fingerprint.

12- Bank (L)

number of permissions =11

This application requires critical permission example :

1. camera (take pictures and videos).
2. Contacts (read, modify)
3. Telephone (read phone status and identity, directly call)
4. Storage (modify, delete , read).

13- Bank (M)

number of permissions =6

This application requires critical permission example :

1. camera (take pictures and videos).
2. Use biometric hardware .

4.2.2 Step two :Evaluation the privacy:

Table 4:1 (result of evaluation privacy)

Bank	Permission								
	Camera	Contacts	Location	Telephone	Storage	Calendar	Bluetooth	biometric	number
A	✓	×	✓	✓	✓	✓	✓	✓	19
B	✓	✓	✓	✓	✓	×	✓	×	17
C	✓	×	✓	×	✓	×	×	×	7
D	✓	✓	✓	✓	✓	×	×	×	11
E	✓	×	✓	✓	✓	×	×	×	11
F	✓	×	✓	✓	✓	×	×	×	9
G	✓	✓	×	×	×	×	×	×	5
H	✓	✓	×	×	✓	×	×	×	7
I	✓	✓	×	×	×	×	×	×	5
J	×	×	×	×	×	×	×	×	2
K	✓	✓	✓	×	✓	×	×	✓	15
L	✓	✓	×	✓	✓	×	×	×	14
M	✓	×	×	×	×	×	×	✓	6

Analysis and results:

- Most of the applications requested access to the camera, and this is normal because it is used to read QR code.
- It is suspicious that applications request access to contacts, calendar and bluetooth although it was found that 50% of the applications requested access to them.
- Some applications use the location feature to search for bank branches or ATMs, and this requires access to the geographical location.
- Some applications request sensitive data such as phone number,IMEI . Is this data stored securely?
- It is normal to ask to write, modify and delete some data.
- Bank (y) Y first in terms of privacy, because it is a request and connection to the Internet only.
- Bank (A)that violates the privacy due to the large number of permission that it requests.

4.3 Securing App Code & Reveres Engineering :

APK format :

The **APK,** ***Android Application Package,*** a bundle is the format used to package any application you develop or that you can get from Google Play Store or any other channel.

An APK file is essentially a ZIP file, so you can get one, rename it and then extract it to have access to its content.

reversing engineering mobile banking in Sudan:

4.3.1Step one testing :

1- bank [A] :

Good app against reverser engineering because cannot recompile

2- Bank (B) :

Good app against reverser engineering because cannot recompile

3- Bank (c) :

Good app against reverser engineering because cannot recompile

4- Bank [D] :

Ability to change :

1. Change of app name
2. All image in client side
3. Change the background
4. Change the all text saved in client side
5. Change the color
6. Change the font type and the style
7. Change the package name and version

Example:

Change Contact us

5- Bank [e]:

Ability to change :

1. Change of app name
2. All image in client side
3. Change the background
4. Change the all text saved in client side
5. Change the color

6. Change the font type and the style
7. Change the package name and version
8. discovering and understanding java code

Example:

Change the background

6- Bank [F]

Good app against reverser engineering because cannot recompile

7- Bank [G]

Good app against reverser engineering because cannot recompile

8- Bank [H]

Ability to change :

1. Change of app name
2. All image in client side
3. Change the background
4. Change the all text saved in client side
5. Change the color
6. Change the font type and the style
7. Change the package name and version
8. discovering and understanding java code

example :

Change the image

9- Bank [I]

Ability to change :

1. Change of app name
2. All image in client side
3. Change the background
4. Change the all text saved in client side
5. Change the color
6. Change the font type and the style
7. Show sources code (java): SqLite Database
8. Change hyper link

Example:

Change hyper link

Show sources code (java): SqLite Database

10- Bank[J] :

Ability to change :

1. Change the design
2. Change the font type and the style
3. Change the HTML file
4. Chanhe the value
5. Change the package name and version

Example :

Change the design login page (password felid):

11- Bank (K)

Good app against reverser engineering because cannot recompile

12- Bank (L)

Ability to change :

1. Change the design
2. Change Direction of transfare data
3. Change the HTML file
4. Chanhe the value
5. Change the package name and version

Example :

Phishing attack generate IPIN "man in the middle attack"

13- Bank (M)

Good app against reverser engineering because cannot recompile

4.3.2 Step two Evaluation the revere engineering**:**

Table 4:2 (result of evaluation reveres engineering)

Bank	Load data from server	Recompile	Evaluation
A	All sensitive data	No	6
B	Some data	No	5
C	All sensitive data	No	6
D	Some data	Yes	1
E	All sensitive data	Yes	2
F	All sensitive data	No	6
G	Some data	No	5
H	Some data	Yes	2
I	Some data	Yes	2
J	Some data	Yes	1
K	Some data	NO	5
L	Some data	Yes	1
M	All sensitive data	No	6

Analysis and results:

- Load data from server very good idea against reveres engineering because can't Edit the data .
- 50% of the applications can be rebuilt after modification.
- The risk of modification is the amount of data that can be modified.
- Changing the data path is one of the most dangerous modifications of applications
- There are 3 mobile banking used the same application with change images and names .

What are the suggested solutions?

1. Do not use local storage retrieve the content from server
2. Save the resource in server side
3. Use the firebase
4. Secure the code using C++ shared libraries .with ProGuard

4.4 Multi – Factor Authentication:

is the name for an authentication method that relies on more than one factor when determining whether to grant access to a computer user. It has become an increasingly important means of proving identity and securing information.

4.4.1 Step one testing:

What is requirement to login mobile banking in Sudan :

1- Bank (A):

Requirement:

1. Username(characters).
2. Password (8 or more characters, upper & lower case letter, at least one number)

2- Bank (B) :

Requirement:

1. Account number(only number)
2. Password(8 or more characters, upper & lower case letter, at least one number)
3. Device ID (imei)

Bank B use 2 factor authentication show this message change the device or imei

Bypass authentication :

The device verification process can be bypassed by changing the imei number with the previously registered number...

3- Bank (C) :

Requirement:

1. Username(characters).
2. Password (8 or more characters, upper & lower case letter)

4- Bank (D) :

Requirement:

1. Username(characters).
2. Password (8 or more characters, upper & lower case letter, at least one number)

5- Bank (E) :

Requirement:

1. Username(characters).
2. Password (8 or more characters, upper & lower case letter, at least one number)

6- Bank (F) :

Requirement:

1. user ID (only number)
2. password(only number 4 digits)

7- Bank (G) :

Requirement:

1. username
2. password

- Create the database on the device
- Verification offline
- Use the IPIN 4 digits

8- Bank (H) :

Requirement:

1. username
2. password

- Create the database on the device
- Verification offline
- Use the IPIN 4 digits

9- Bank (I) :

Requirement:

1. username
2. password

- Create the database on the device
- Verification offline
- Use the IPIN 4 digits

10- Bank (J) :

Requirement:

1. Phone Number
2. IPIN (only number)

11- Bank (K)

Requirement:

1. Phone number
2. Password (8 or more characters, upper & lower case letter, at least one number)
3. Fingerprint.

12- Bank (L)

Requirement:

1. Username (character)
2. Password (more characters, upper & lower case letter, at least one number).

13- Bank (M)

Requirement:

1. Phone number
2. Password (8 or more characters, upper & lower case letter, at least one number).

4.4.2 Step two :Evaluation factor authentication:

Table 4:3 (result of factor authentication)

Bank	Factor Authentication				
	Username	Password	Device ID	fingerprint	Evaluation
A	✓	✓	×	✓	6
B	✓	✓	✓	×	5
C	✓	✓	×	×	4
D	✓	✓	×	×	4
E	✓	✓	×	×	4
F	✓	✓	×	×	2
G	✓	✓	×	×	1
H	✓	✓	×	×	1
I	✓	✓	×	×	1
J	✓	✓	×	×	1
K	✓	✓	×	✓	5
L	✓	✓	×	×	4
M	✓	✓	×	✓	6

Analysis and results:

- Using more than one Authentication factor increases security
- 40% of apps do not have security policies when creating the password
- Some applications allow you to create a 4-digit password
- The four-digit password can be guessed using social engineering.
- Using upper and lower case letters, numbers, and symbols when creating a password reduces the possibility of guessing the password

4.5 Secure Communication & Encryption Transaction :

Secure communication is when two entities are communicating and do not want a third party to listen in.

4.5.1 step one : Check the security communication mobile banking in Sudan

1- Bank (A) Destnation ip =196.1.218.250

- Secure connection between client and server .
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2

2- Bank (B) Destination ip= 196.29.170.119

- Secure connection between client and server .
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2

3- Bank (C) Destination ip = 196.202.140.100

- unsecure connection between client and server .
- Transaction with out encryption .
- Transport layer http.
- username & password clear over the http.

4- Bank(D) destination ip= 212.0.146.75:

- Secure connection between client and server
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.

5- Bank (E) Destination ip 196.223.157.100

- unsecure connection between client and server .
- Transaction with out encryption .
- Transport layer http.
- username & password clear over the http.

6- Bank (F) Destination ip =212.0.146.46

- Secure connection between client and server
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.

7- Bank (G) destination ip = 212.0.132.14

- Create the database on the device

- Verification offline
- Use the IPIN 4 digits
- Secure connection
- Transport layer security TLS V1.2

8- Bank (H) Destination ip = 212.0.132.14

- Create the database on the device
- Verification offline
- Use the IPIN 4 digits
- Secure connection
- Transport layer security TLS V1.2

9- Bank (I) destination ip = 212.0.132.14

- Create the database on the device
- Verification offline
- Use the IPIN 4 digits.
- Secure connection.
- Transport layer security TLS V1.2

10- Bank (J) destination ip = 212.0.140.44

- Secure connection between client and server
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.

11- Bank (K) destination ip=51.77.12.37:

- Secure connection between client and server
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.
-

12- Bank (L)destination ip=212.0.150.117:

- Secure connection between client and server
- Encryption transaction

- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.

13- Bank (M)destination ip=212.0.150.117:

- Secure connection between client and server
- Encryption transaction
- Uses the EC Diffie-hellman algorithm for Key exchange
- Transport layer security TLS V1.2.

4.5.2 step two: Evaluation the communication:

Table 4:4 (Result of the connection check)

Bank	Destination IP	Protocol	Encryption	Evaluation
A	196.1.218.250	TLS 1.2	Yes	6
B	212.0.144.193	TLS 1.2	YES	6
C	196.202.140.100	HTTP	NO	1
D	216.58.208.142	TLS 1.2	YES	6
E	196.233.157.100	HTTP	NO	1
F	197.254.204.202	TLS V1.2	YES	6
G	212.0.132.14	TLS V1.2	YES	3
H	212.0.132.14	TLS V1.2	YES	3
I	212.0.132.14	TLS V1.2	YES	3
J	212.0.140.44	TLS V1.2	YES	4
K	51.77.12.37	TLS V1.2	YES	5
L	212.0.150.117	TLS V1.2	YES	5
M	212.0.148.125	TLS V1.2	yes	5

Analysis and results:

- More than 80% applications use the secure communication between client and server.
- Username and password are not encrypted in unsecured communication .
- The password can be obtained when you are on the same insecure network .
- There are 3 mobile banking used the same server .

4.6 Encryption data storage:

Sometime android developers stores sensitive information without encryption .

Application might stores data in shared preferences:

- ✓ Data / data /[packagename]/shared_preferences
- ✓ Database
- ✓ Temporary files
- ✓ External storage

4.6.1 Step one Check the security data storage in Mobile banking :

1- Bank (A) not work now :

- No database found
-

2- Bank(B) Mbok

- database :

no database found

- file :

User data storage without encryption:

1. Serial number mobile imei
2. Account number
3. Path = data/data/ com.mode.bok.ui / IpbPrefs.xml

3- Bank (C):

- Database:
  NO database found " temporary data "
- File :

Not sensitive data found

4- Bank(D)

- database :

no database found

- file :

  User data storage without encryption:

  1. account name , last login,
  2. Account number
  3. Path = data/data/ a2a.android.onb/sharedPrefs

5- Bank (E) :

- Database:

No database found

- File :

Not sensitive data found

6- Bank (F) :
Good idea the app not work in root case

- Database :

no database found

- File :

  User data storage without encryption:

  1. Account Name
  2. Account number
  3. Path = data/data/ com.mode.fib.ui/ LoginPrefs.xml

7- Bank (G) Abshare

- Database: name ‘fav_lists.db’
  Table:
  1. User_cards_table
  2. User_table
- File:

  User data storage without encryption:

  1. Account Name
  2. phone number
  3. Path = data/data/ com.aztech.abshir/sharedprefs/ PrefFile.xml

8- Bank (H):

- Database: name ‘fav_lists.db’
  Table:
    1. user_cards_table
    2. user_table

- File:

User data storage without encryption:

1. phone number
2. Path = data/data/ com.aztech.ARB/sharedprefs/ PrefFile.xml

9- Bank (I):

- Database: name ‘fav_lists.db’
  Table:
    1. user_cards_table
    2. user_table

- File:

User data storage without encryption:

1. phone number
2. Path = data/data/ com.saudibank.mobile/sharedprefs/ PrefFile.xml

10- Bank (J):

11- Bank (K):

- Database: name ‘syberapp’
  Table:
    1. Cards

    2. users

- file :

User data storage without encryption:

1.code applock, phone number

2. android version ,sdk ….

3. Path = data/data/com.sybertechnology.syberapp.live.release/sharedPrefs

12- Bank (L)

- database name “pancard” :

- file :

  User data storage without encryption:

  1. account name , exptdata
  2. card number
  3. Path = data/data/ a2a.android.onb/sharedPrefs

13- Bank (M)

- database :

no found

- file :

  User data storage encrypted:

  1. account name , exptdata
  2. card number
  3. Path = data/data/ com.busra.pay/sharedPrefs

Step two Evaluation Data storage :

Table 4:5 (**Result of data storage check**)

Result of data storage check :

Bank	Database	Sensitive data	Encryption	Evaluation
A	Not found			
B	Not found	Found	No	5
C	Not found	Not found	-	6
D	Not found	Found	No	4
E	Not found	Not found	-	6
F	Not found	Found	No	5
G	Found	Yes	Something	1
H	Found	Yes	Something	1
I	Found	Yes	Something	1
J	N/A	N/A	N/A	
K	Found	Yes	Something	1
L	Found	Yes	Something	1
M	Not found	Not found	Yes	6

Analysis and results:

- Storing data on the phone helps in speedy response Storing and reduce the load on the service provider
- More than 50% of applications store sensitive data insecurely.
- Some applications create databases on the phone.
- Some applications use encryption when storing data on the phone.

4.7 Session Time Out :

A session is the occurrence of a set of user interactions with a web application during a specific time frame. For example, a single session can contain multiple page views, events, social interactions, and e-commerce transactions

4.7.1 Step one testing the Session time out in mobile banking in Sudan :

1- Bank (A)

Not work

2- Bank (B)mbok :

1. Session time out = **5 minutes**

2. **only one** session on real time

3- Bank (C)

1. session time out **= 5 minutes**

2. **more than one session** at the same time

4- Bank (D)

1. Session time out **= 2 minutes**

2. **only one** session in real time

5- Bank (E)

Not available

6- Bank (F)

1. Session time out **=10 minutes**

2. **only one session** at the same time .

7- Bank (G) :

No session

8- Bank (H):

No session

9- Bank (I) :

No session

10- Bank (J)

Not available

11- Bank (K)

1 . Session time out **=5 second**

2. local session

3. **more than one session** at the same time

12- Bank (L):

1. Session time out **=5 seconds**

2. local session

3. more than one session at the same time .

13- bank (M):

. Session time out **=7 minute**

2. local session

3. **more than one session** at the same time

4.7.2Step two :Evaluation the session time out:

Table 4:6 (**Result of session time out**):

Bank	Used Session Time Out	Period	Only One Session at the same time	Evaluation
A	N/A	N/A	N/A	
B	Yes	5 Minutes	Yes	5
C	Yes	5 minutes	NO	4
D	Yes	2 minutes	Yes	6
E	N/A	N/A	N/A	
F	Yes	10 Minutes	Yes	4
G	No	No	No	1
H	No	No	No	1
I	No	No	No	1
J	N/A	N/A	N/A	
K	Yes	5 seconds	No	5
L	Yes	5 seconds	No	5
M	Yes	7 minute	N0	3

Analysis and results:

- Not using the session may lead to theft of money.
- More than 50% application allow creating one or more session at the same time .
- The shorter the session , the more security .
- The larger the session increased load on the server provider.

CHAPTER V

Conclusion and future work

Chapter 5

Conclusion and future work

5-1 Conclusion

- Some applications require more access than is required
- Some applications can be modified and published
- Some applications do not have security policies when creating the password, such as creating a password from 4 fields
- Some applications exchange username and password without encryption
- Some applications allow storing sensitive data on the phone such as card number, phone number ...
- Some applications contain longer session time leading to unauthorized access.

Table 5:1 (**security Evaluation**)

Bank	security Evaluation
A	3.17
B	4.67
C	3.83
D	4.00
E	2.67
F	4.50
G	2.67
H	2.00
I	2.17
J	2.00
K	3.83
L	3.00
M	5.00

5.2 future work :

This area is a good area for researchers and this subject can be studied from many other aspects

1. Applications can be evaluated based on one factor of previous factors with detail.
2. Evaluation application on the ease of use .
3. Evaluation of applications functionality .
4. Fixed standard of the university of Sudan can be measures application and upload the result on web site .

5.3 References:

[1] D.Pogue, A Place to Put Your Apps. New York Times (2013).

[2] ""App" voted 2010 word of the year by the American Dialect Society (UPDATED) American Dialect Society". Americandialect.org. 2011-01-08. Retrieved 2012.

[3] BBVA. Mobile Banking. Centro de Innovación BBVA. p. 22.

[4] Ashok Bahadur Singh,"Mobile banking based money order for India Post: Feasible model and assessing demand potential", International conference on emerging economies-Prospects and challenges (2012)

[5] Jeong, B. K., & Yoon,"An Empirical Investigation on Consumer

Acceptance of Mobile Banking Services", Business and Management Research, 2(1), 31-40, T. E. (2013)

[6] J. D. Pitts, "Surfing the Payment Channels, Mastering the Fraud

Tsunami", JDP Enterprises, Carrollton, TX, (2010).

[7] Balebako, R., & Cranor, L., "Improving App Privacy: Nudging App Developers to Protect User Privacy", Security & Privacy, IEEE, 12(4), 55-58, (2014)

[8] Md. Shoriful Islam," Systematic Literature Review: Security Challenges of Mobile Banking and Payments System", International Journal of uand e- Service, Science and Technology Vol. 7, pp. 107-116(2014)

[9] Elkhodr, M., Shahrestani, S., & Kourouche, K.," A pro-posal to improve the security of mobile banking applications", In ICT and Knowledge Engineering (ICT & Knowledge Engineer-ing), 2012 10th International Conference on (pp. 260-265). IEEE, (2012)

[10] He, W., "A Review of Social Media Security Risks and Mitigation

Techniques", Journal of Systems and Information Technology, 14(2), 171-180, (2012)

[11] He, W. , "A Survey of Security Risks of Mobile Social Media through Blog Mining and an Extensive Literature Search", Information Management and Computer Security, 21(5), pp.381–400.

[12] Paul Judge and Mostafa Ammar, "Security Issues and Solutions in Multicast Content Distribution: A Survey ", IEEE Network,(2003)

[13] Hameed Ullah Khan, "E-banking: Online Transactions and Security Measure", Research Journal of Applied sciences, Engineering and technology 7(19): 4056-4063, (2014)

[14] Rajpreet Kaur Jassal1, Ravinder Kumar Sehgal," Study of Online Banking Security Mechanism in India: Take ICICI Bank as an Example ", IOSR journal of computer engineering (IOSR-JCE), p-ISSN: 2278- 8727 Volum13, Issue1 ,PP114-121,(2013),

[15] Anthony Luvanda," Proposed Framework for Securing Mobile Banking Applications from Man in the Middle Attacks", Journal of Information Engineering and Applications www.iiste.org ISSN 2224-5782 (print) ISSN 2225-0506 (online),Vol.4, (2014).

[16] Kelvin Chikomo, Ming Ki Chong, Alapan Arnab, Andrew Hutchison'Security of Mobile Banking', Department of Computer Science University of Cape Town ,Rondebosch 7701, South Africa

[17] Azamjon Abdullaev , Mohammed Abdulhakim Al-Absi ,Ahmed Abdulhakim Al-Absi , Mangal Sain, Young-Sil Lee, Hoon Jae Lee 'Security Challenge and Issue of Mobile Banking in Republic of Uzbekistan: A State of Art Survey',ICACT2019, (2019).

[18] Alexis Rusagara, Dr. Cheruiyot W.K, Dr. Anthony Luvanda "Securing Online Banking Services against Man in the Middle Attacks by use of two Factor Authentication",IJISET, Vol. 3 Issue (2016).

[19] Kemal Bicakcia, Devrim Unalb, Nadir Ascioglucc, Oktay Adalierc,'Mobile Authentication Secure Against Man-In-The-Middle Attacks' Kocaeli 41470, Turkey,MobiSPC-(2014).

[20] A joint research between Inside Secure and UL'THE WILD WEST OF MOBILE SECURITY '(2018).

[21] Pablo F. Ordo˜ nez-Ordo˜ nez1,2(B) , Domingo D. Herrera-Loaiza1 , and Roberth Figueroa-Diaz 'Vulnerabilities in Banking Transactions with Mobile Devices Android: A Systematic Literature Review'Springer Nature Switzerland (2019).

[22] https://dictionary.cambridge.org/dictionary/english/privacy ,3-2-2020, 4:00pm.

[23]https://matomo.org/blog/2014/01/data-privacy-day-january-28th/,6-2-2020, 4:00pm.

[24] https://developer.android.com/privacy,1-2-2020, 10:00pm.

[25] https://www.apple.com/privacy/control/,9-2-2020, 4:00pm.

[26]https://www.telesign.com/resources/research-and reports/telesign-consumer-account-security-report/,7-4-2020, 10:00pm.

[27] "Two-factor authentication: What you need to know (FAQ) – CNET". *CNET*. Retrieved 2015.

[28] https://en.wikipedia.org/wiki/Communications_security ,20-1-2020, 8:00pm.

[29] https://www.thalesesecurity.com/faq/encryption/what-storage-encryption

[30]https://www.esecurityplanet.com/cloud/data-storage-security.html ,11-2-2020, 4:00pm.

[31] https://owasp.org/www-community/Session_Timeout ,2-3-2020, 7:00pm.

[32] https://kb.blackbaud.com/articles/Article/64114 ,3-5-2020, 1:00pm.