Università di Pisa

Dependability and Secure System 2025

Andrea Mugnai

Jacopo Tucci

University of Pisa

Introduction

Static Analysis

Starting from VirusTotal analysis and integrating it with MobSF we classified our samples as:

- **FakeBank** (4 samples)
- **RansomLock** (1 sample)

The initial classification made by those tools was subsequently verified by analyzing the source code of the APKs decompiled using **JD-GUI**.

Dynamic Analysis

The dynamic analysis was performed using the **Android Emulator** provided by **Android Studio**. Simulating a Pixel XL device with **Android 9.0** (API level 28), we installed and executed the APKs using the **MobSF Dynamic Analysis** tool in a **Virtual Machine**.

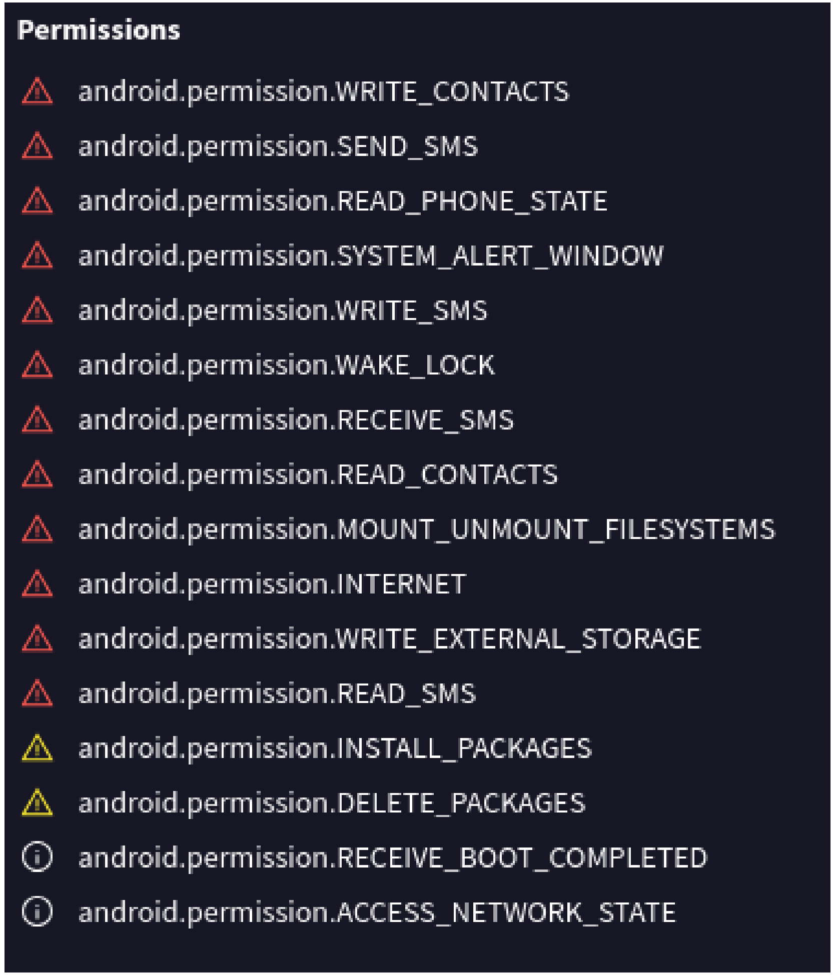
Fakebank Family

Fakebank Family

IBK
IBK
KB국민카드
신한은행

The four analyzed **Malware** samples are **Trojan bankers** designed to mimic legitimate banking apps. In reality, they steal sensitive user information such as phone numbers and banking credentials. Additionally, they intercept all incoming SMS messages to capture one-time passwords `(OTPs)` sent by the bank.

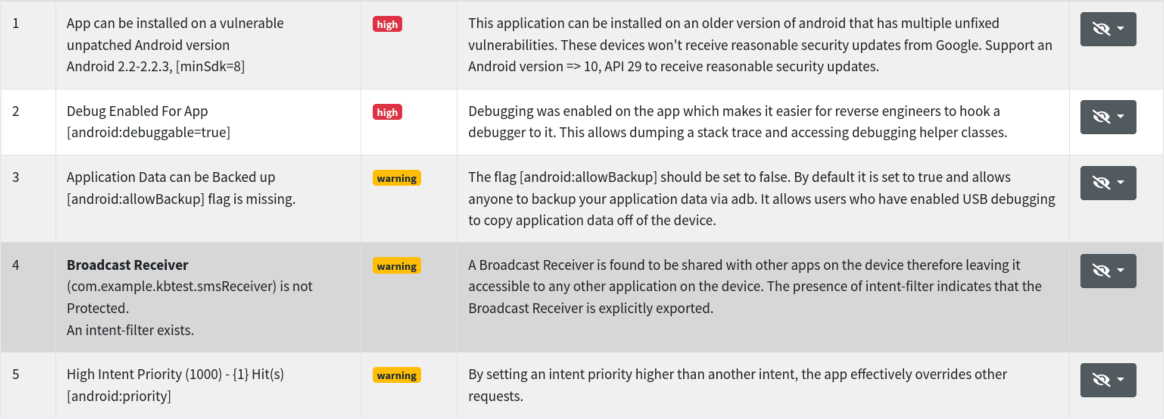
Permissions



Permissions

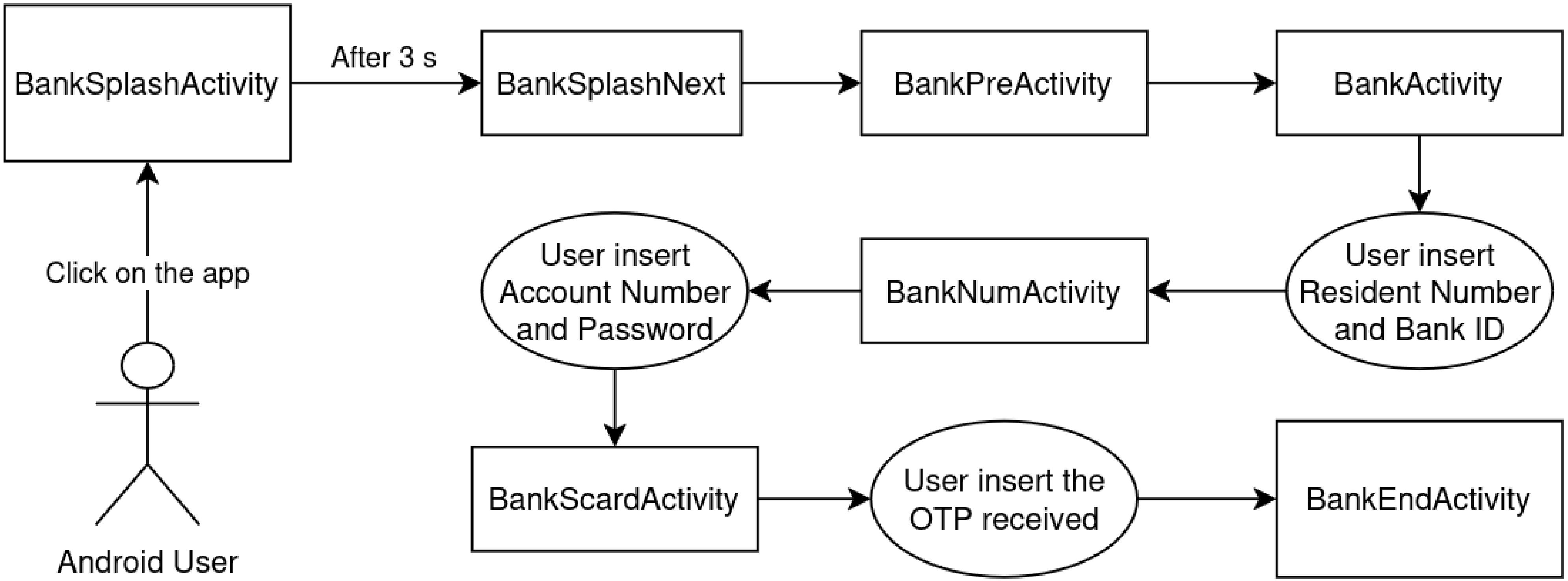
- ⚠ android.permission.WRITE_CONTACTS
- ⚠ android.permission.SEND_SMS
- ⚠ android.permission.READ_PHONE_STATE
- ⚠ android.permission.SYSTEM_ALERT_WINDOW
- ⚠ android.permission.WRITE_SMS
- ⚠ android.permission.WAKE_LOCK
- ⚠ android.permission.RECEIVE_SMS
- ⚠ android.permission.READ_CONTACTS
- ⚠ android.permission.MOUNT_UNMOUNT_FILESYSTEMS
- ⚠ android.permission.INTERNET
- ⚠ android.permission.WRITE_EXTERNAL_STORAGE
- ⚠ android.permission.READ_SMS
- ⚠ android.permission.INSTALL_PACKAGES
- ⚠ android.permission.DELETE_PACKAGES
- ⓘ android.permission.RECEIVE_BOOT_COMPLETED
- ⓘ android.permission.ACCESS_NETWORK_STATE

The most dangerous permissions used by the malware are those related to managing SMS messages and phone calls. Combined with the **broadcasting** permission and by assigning the app the highest priority (**1000**), the malware is able to intercept all incoming SMS messages and phone calls.



1	App can be installed on a vulnerable unpatched Android version Android 2.2-2.2.3, [minSdk=8]	high	This application can be installed on an older version of android that has multiple unfixed vulnerabilities. These devices won't receive reasonable security updates from Google. Support an Android version => 10, API 29 to receive reasonable security updates.
2	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.
3	Application Data can be Backed up [android:allowBackup] flag is missing.	warning	The flag [android:allowBackup] should be set to false. By default it is set to true and allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.
4	**Broadcast Receiver** (com.example.kbtest.smsReceiver) is not Protected. An intent-filter exists.	warning	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported.
5	High Intent Priority (1000) - {1} Hit(s) [android:priority]	warning	By setting an intent priority higher than another intent, the app effectively overrides other requests.

App Flow



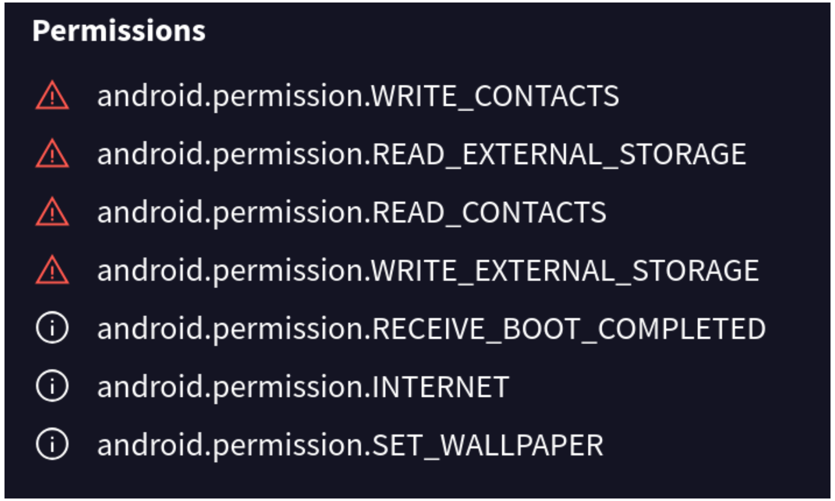
`BankSplashActivity` and `BankEndActivity` sends all the stolen data to a remote server that is not the legitimate one. (`http://banking1.kakatt.net:9998`)

RansomLock Family

RansomLock Family

The **Malware** sample is a **Locker Ransomware** that imitate a famous strategy game. Actually, it encypt both file contents and contact data like a typical ransomware, while launches a game-themed fake lock screen on every reboot—hiding its icon and blocking any escape until the ransom is paid.

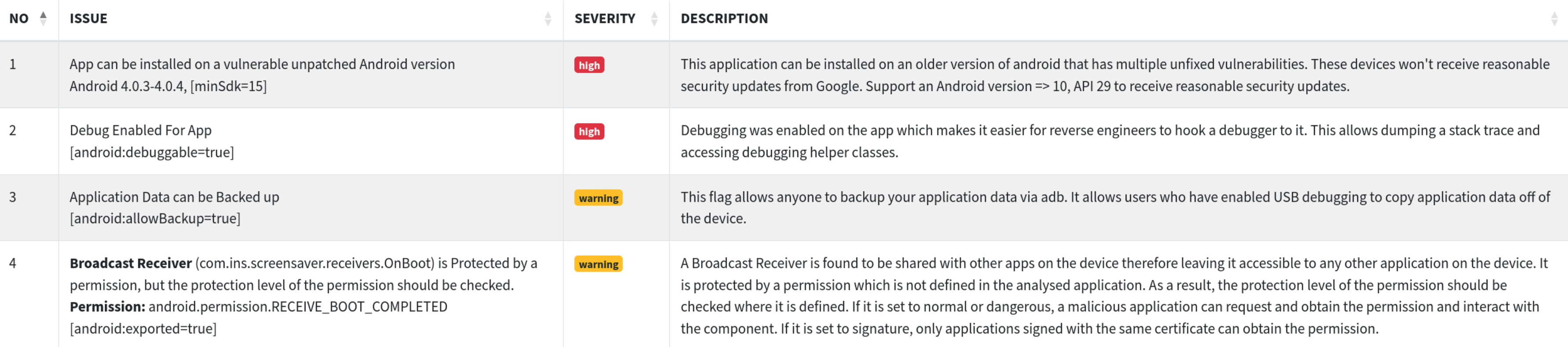
Permissions



Permissions

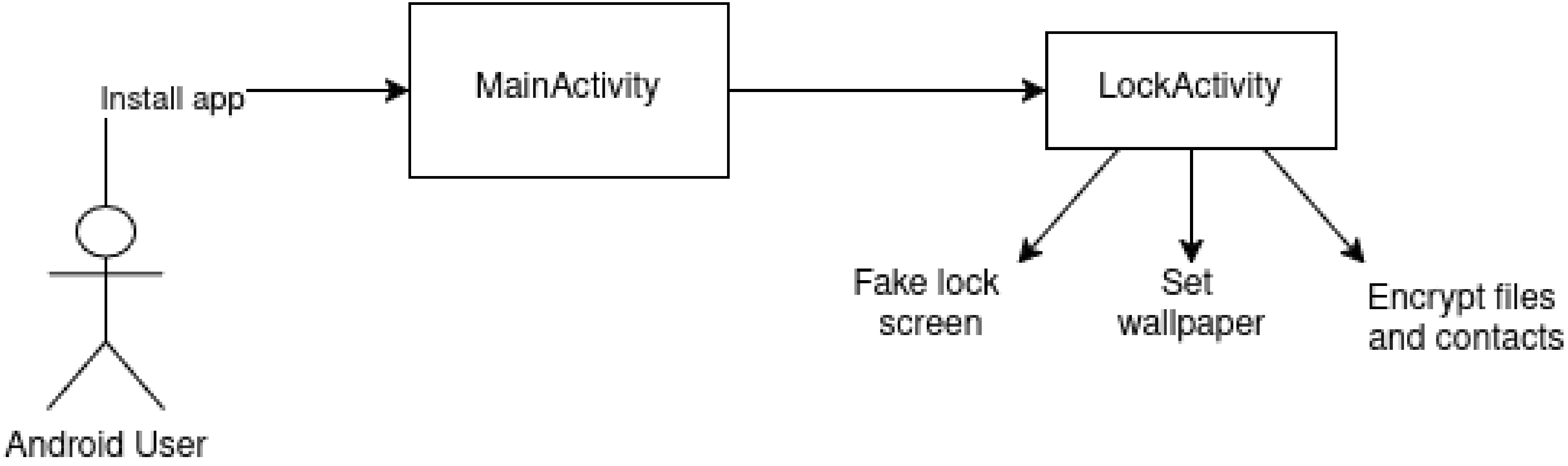
- ⚠ android.permission.WRITE_CONTACTS
- ⚠ android.permission.READ_EXTERNAL_STORAGE
- ⚠ android.permission.READ_CONTACTS
- ⚠ android.permission.WRITE_EXTERNAL_STORAGE
- ⓘ android.permission.RECEIVE_BOOT_COMPLETED
- ⓘ android.permission.INTERNET
- ⓘ android.permission.SET_WALLPAPER

The sample requests dangerous permissions to read and write external storage and contacts in order to encrypt personal data. Using the RECEIVE_BOOT_COMPLETED permission and an exported **BroadcastReceiver**, it auto starts at boot.



NO	ISSUE	SEVERITY	DESCRIPTION
1	App can be installed on a vulnerable unpatched Android version Android 4.0.3-4.0.4, [minSdk=15]	high	This application can be installed on an older version of android that has multiple unfixed vulnerabilities. These devices won't receive reasonable security updates from Google. Support an Android version => 10, API 29 to receive reasonable security updates.
2	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.
3	Application Data can be Backed up [android:allowBackup=true]	warning	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.
4	**Broadcast Receiver** (com.ins.screensaver.receivers.OnBoot) is Protected by a permission, but the protection level of the permission should be checked. **Permission:** android.permission.RECEIVE_BOOT_COMPLETED [android:exported=true]	warning	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analysed application. As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission.

App Flow



`LockActivity` requests a symmetric key from a remote server to encrypt files, and subsequently decrypts them if the payment is made. (`http://timei2260.myjino.ru`)