Exploiting OLA Money v1.9

Summary: The app has the following weaknesses which can be exploted to compromise the app completely - Credentials Leak, Authentication Bypass using Alternate Channel, n is possible on OLA Money app. The bug was responsibly disclosed and the vendor has indicated that the vulnerability is not severe enough.

Vendor Website:https://www.olacabs.com/whiteh

Basic Details

Package: com.olacabs.olamoney
Application Label: Ola Money
Process Name: com.olacabs.olamoney
Version: 1.9.0
Data Directory: /data/data/com.olacabs.olamoney
APK Path: /data/app/com.olacabs.olamoney-1.apk
UID: 10124
GID: [3003]
Shared Libraries: null
Shared User ID: null
Uses Permissions:

- android.permission.ACCESS_NETWORK_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.INTERNET
- android.permission.RECEIVE_SMS
- android.permission.SEND_SMS
- android.permission.READ_CONTACTS
- android.permission.READ_PHONE_STATE
- android.permission.VIBRATE
- android.permission.CAMERA
- android.permission.ACCESS_FINE_LOCATION
- com.google.android.c2dm.permission.RECEIVE
- android.permission.ACCESS_COARSE_LOCATION
- android.permission.READ_SMS
- com.ola.sdk.device.platform.permission.MQTT_RECEIVE_MESSAGE_BROADCAST
- android.permission.WAKE_LOCK
- com.olacabs.olamoney.permission.C2D_MESSAGE

Activities exported: 2

- com.olacabs.olamoney.activities.SplashActivity
  - Permission: null
- com.olacabs.olamoneyrest.core.PayActivity
  - Permission: null

Broadcast receivers exported: 6

- com.apsalar.sdk.ApsalarReceiver
  - Permission: null
- com.olacabs.olamoney.receivers.NetworkChangeBroadcastReceiver
  - Permission: null
- com.google.android.gms.gcm.GcmReceiver
  - Permission: com.google.android.c2dm.permission.SEND
- com.ola.sdk.deviceplatform.InternalReceiver
  - Permission:*com.ola.sdk.deviceplatform.permission.MQTT_RECEIVE_MESSAGE_BROADCAST*
- com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver
  - Permission: android.permission.INSTALL_PACKAGES
- com.google.firebase.iid.FirebaseInstanceIdReceiver
  - Permission: com.google.android.c2dm.permission.SEND

Content providers exported: 0
Services exported: 2

- com.olacabs.batcher.OffTimeService
  - Permission: com.google.android.gms.permission.BIND_NETWORK_TASK_SERVICE
- com.google.firebase.iid.FirebaseInstanceIdService
  - Permission: null

Work flow diagram of Ola Money

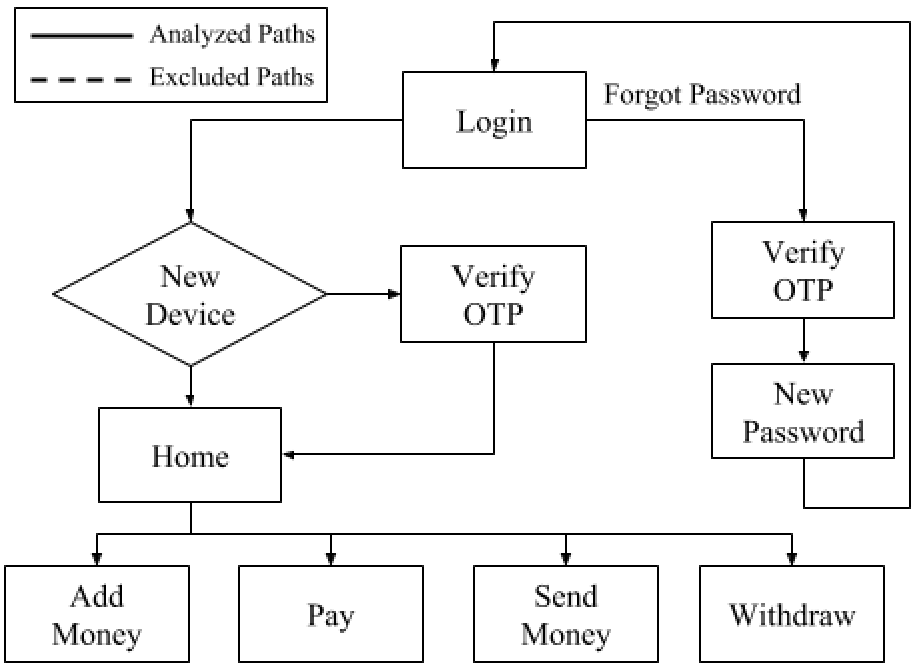
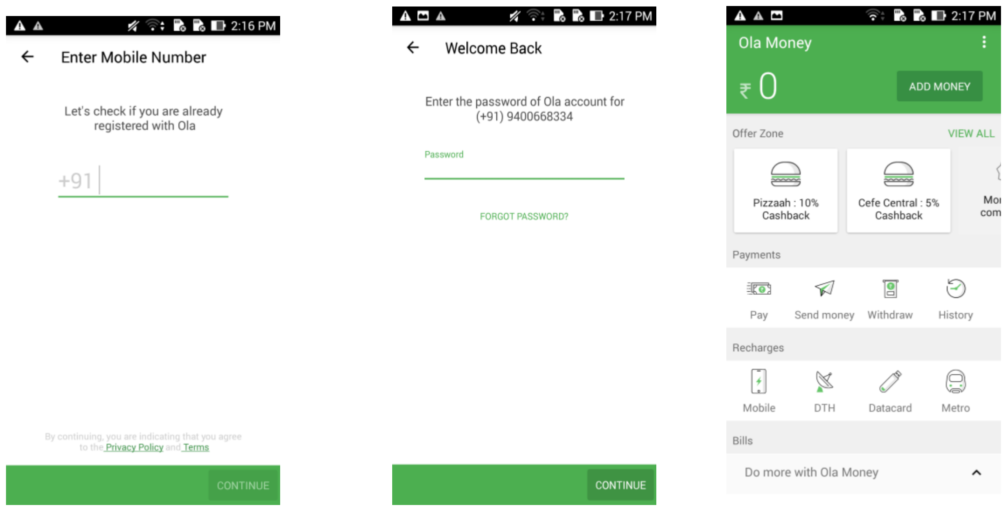


Fig : Workflow of Ola Money

Overview of Ola Money Authentication System

The screenshots below show the normal authentication system of OLA Wallet.



Let's check if you are already registered with Ola

By continuing, you are indicating that you agree to the Privacy Policy and Terms

Enter the password of Ola account for (+91) 9400668334

Fig: Authentication in Ola

Setup & Assumptions

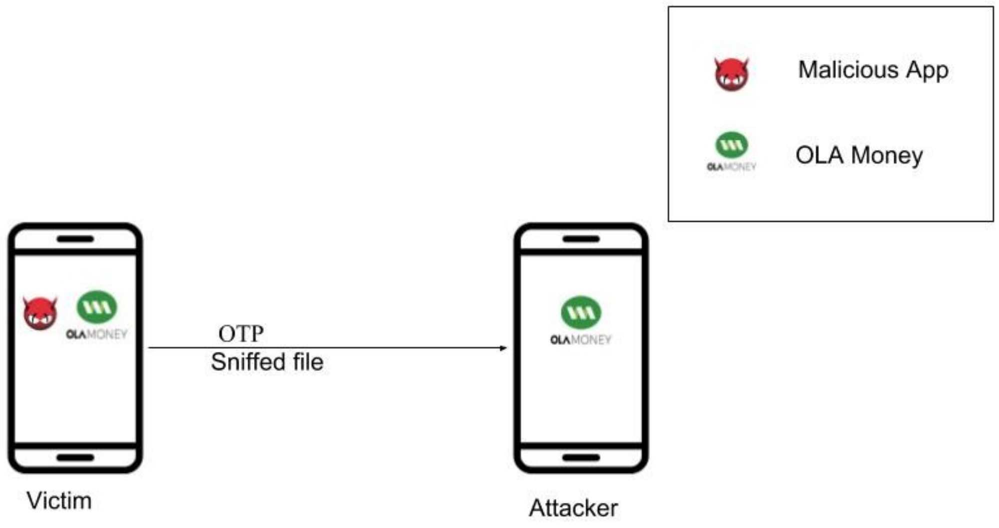
The attack involves a victim device and an attacker device, both of which have the version of Ola Money app from Google Play installed. We assume that the user has gone through the device authentication process that happens for a first time user.

At some point the device has been compromised resulting in a malicious app running in the background. The malicious app has the capability to:

1. Intercept SMS messages.
2. Sniff and log user name,password,card number into a file.
3. Forward OTP and password to attacker.

The Exploitation

The malicious app makes use of the accessibility permission along with permission to read and send SMS. This malicious application reads OTP from the messages and sends it to the attacker's phone as SMS. It also logs the user input from the application into the file.



Vulnerability Findings

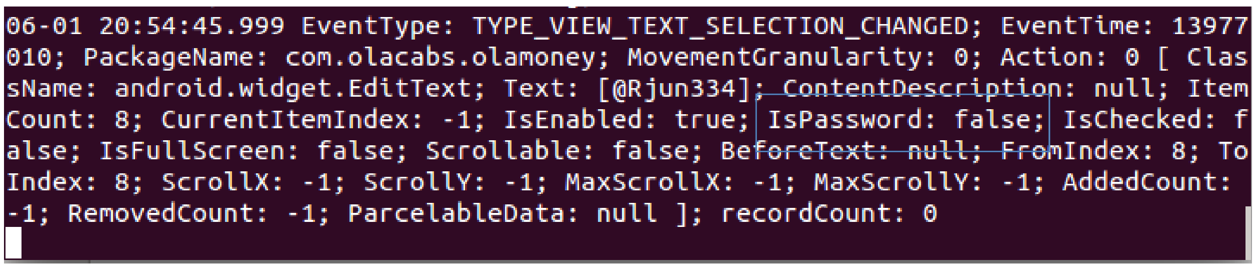
1. **Information Exposure**

Description :

Information exposure is the intentional or unintentional disclosure of information to an actor that is not explicitly authorised to have access to that information

Exploitation Findings

Ola Money login screen has username and password as input textboxes. For username attribute, the input textbox of the keypad is categorized as “not password” which causes the input to be unmasked. Masking the input is enabled using an *isPassword* flag on the textbox. This enables the input to be sniffed through accessibility service. The password field is masked by default. However, for usability purposes, the password field is supported by a show/hide option which the user can toggle. If user clicks on "Show" password then the isPassword attribute on the textbox is set to *false* causing the password to be unmasked. This will allow the malicious app to sniff the password. This is a classic example of a scenario where usability jeopardizes security.



```
06-01 20:54:45.999 EventType: TYPE_VIEW_TEXT_SELECTION_CHANGED; EventTime: 13977
010; PackageName: com.olacabs.olamoney; MovementGranularity: 0; Action: 0 [ Clas
sName: android.widget.EditText; Text: [@Rjun334]; ContentDescription: null; Item
Count: 8; CurrentItemIndex: -1; IsEnabled: true; IsPassword: false; IsChecked: f
alse; IsFullScreen: false; Scrollable: false; BeforeText: null; FromIndex: 8; To
Index: 8; ScrollX: -1; ScrollY: -1; MaxScrollX: -1; MaxScrollY: -1; AddedCount:
-1; RemovedCount: -1; ParcelableData: null ]; recordCount: 0
```

Disclosure: OLA does not acknowledge this as an issue stating usability as a reason.

Exploit:
1. Create an Trojan app that has permission to use accessibility service.
2. The accessibility service intercepts all events and data (sensitive or not) from Ola Money app.

Remediation:
While Show/Hide option on a webpage has not security implications, this has serious implications on an app. The feature must not be supported unless Android framework has some alternative solution for the same. We recommend removing the Show/Hide option for the password field

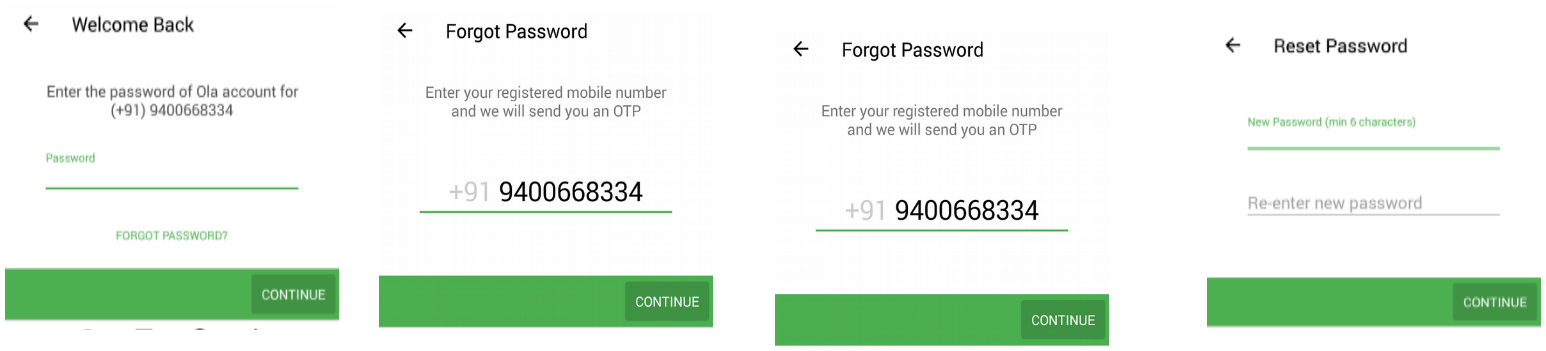
2. Authentication Bypass Using an Alternate Path or Channel

Description:
A product requires authentication, but the product has an alternate path or channel that does not require authentication

Exploitation Findings

An alternate channel i.e, the “Forgot Password” mechanism is used to bypass authentication. When instead of entering the password, the user clicks on “Forgot Password”, the user is taken to a screen where he is prompted to enter the mobile number. At this point, the user has to enter the registered mobile number. This maybe an attacker number or a user’s own registered number. We enter the user’s registered mobile number. An OTP is sent to the mobile number, which is captured by the malicious app and sent to the attacker. The attacker can use it to reset the password. The device is not authenticated prior to resetting the password. .

Fig: Forget pin Workflow



Welcome Back

Enter the password of Ola account for (+91) 9400668334

Password

FORGOT PASSWORD?

CONTINUE

Forgot Password

Enter your registered mobile number and we will send you an OTP

+91 9400668334

CONTINUE

Forgot Password

Enter your registered mobile number and we will send you an OTP

+91 9400668334

CONTINUE

Reset Password

New Password (min 6 characters)

Re-enter new password

CONTINUE

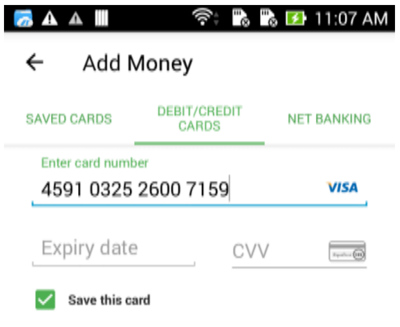
Remediation:
Proper validation of source of OTP message.

3. Exposure of Private Information ('Privacy Violation')

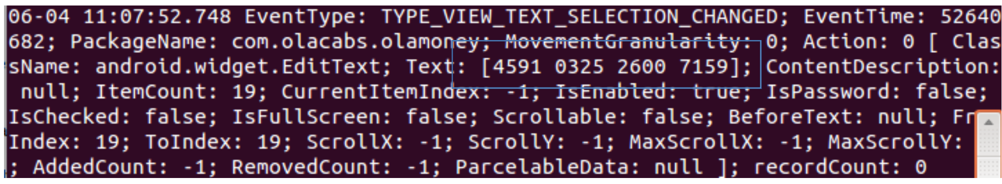
Description:
Software does not properly prevent private data from being accessed by actors who are not authorised to access the data

Exploit Findings
The same malicious app is used for this exploit. The following information can be leaked:

- Credit Card number
- Expiration Date
- Bank Account No.
- Transaction history



Enter card number
4591 0325 2600 7159
Expiry date
CVV



```
06-04 11:07:52.748 EventType: TYPE_VIEW_TEXT_SELECTION_CHANGED; EventTime: 52640
682; PackageName: com.olacabs.olamoney; MovementGranularity: 0; Action: 0 [ Clas
sName: android.widget.EditText; Text: [4591 0325 2600 7159]; ContentDescription:
 null; ItemCount: 19; CurrentItemIndex: -1; IsEnabled: true; IsPassword: false;
IsChecked: false; IsFullScreen: false; Scrollable: false; BeforeText: null; Fr
Index: 19; ToIndex: 19; ScrollX: -1; ScrollY: -1; MaxScrollX: -1; MaxScrollY:
; AddedCount: -1; RemovedCount: -1; ParcelableData: null ]; recordCount: 0
```

Fig : Add Money Screen of Ola Money

Remediation:

Sensitive data should be hidden from other apps.

4. **Improper Ownership Management**

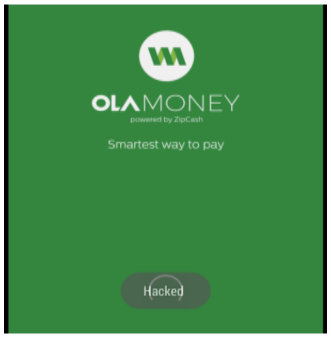
Description

The software assigns the wrong ownership, or does not properly verify the ownership, of an object or resource.

Exploit

A repackaged app modifies the ownership of the app. Table below shows the signatures of the original app and the repackaged app. Ola Money app does not check for the authenticity of the app. The reoackaged app will run on the device without any warning

Signature of Ola Money	Signature of Repackaged App
Signature Algorithm: sha256WithRSAEncryption Issuer: CN=Ajinkya Potdar Validity Not Before: Jun 2 03:28:49 2012 GMT Not After : Oct 4 03:28:49 3011 GMT Public Key Algorithm: rsaEncryption Public-Key: (2048 bit)	Signature Algorithm: sha256WithRSAEncryption Issuer: C=Unknown, ST=Unknown, L=Unknown, O=Unknown, OU=Unknown, CN=Unknown Validity Not Before: Jun 4 08:31:32 2018 GMT Not After : Oct 20 08:31:32 2045 GMT Public Key Algorithm: rsaEncryption Public-Key: (2048 bit)



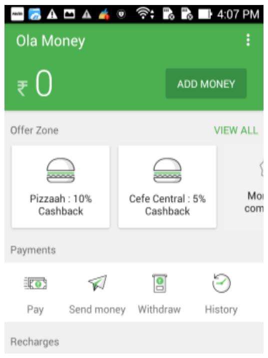


Fig Repackaged application

Remediation:

Proper Implementation of tamper Detection Techniques