Reverse Engineering

Smartphone app:
Fietsknooppunten

Anouk Brondijk

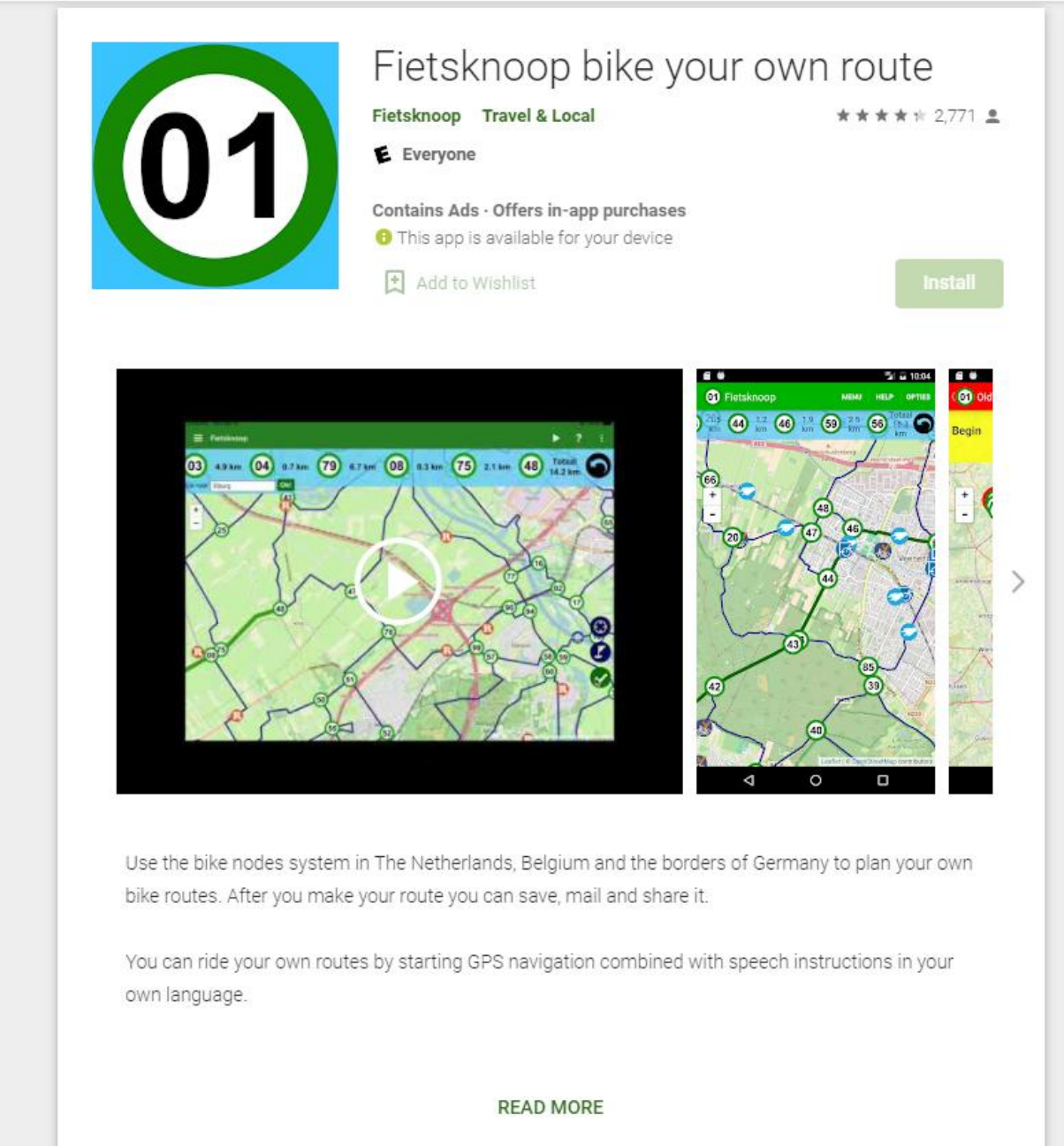
Contents

1. The App 2
   a. About the App 2
   b. Technical Information 2
2. Reversing Process 3
   Reversing the Application 3
3. Findings 3
   Interface + Base Activity 3
   POST – request 7
   Password usage and storage 9
Conclusion 10

1. The App

a. About the App

Fietsknooppunten is an app that lets you choose the route for your next bicycle trip. You can log in, save routes, find friends, know your location, and more.



b. Technical Information

The current version of the app is 4.8.2. It requires Android 6.0 and up. It was last updated December 23, 2020, and there have been no new updates. I will be working in version 4.8.2.

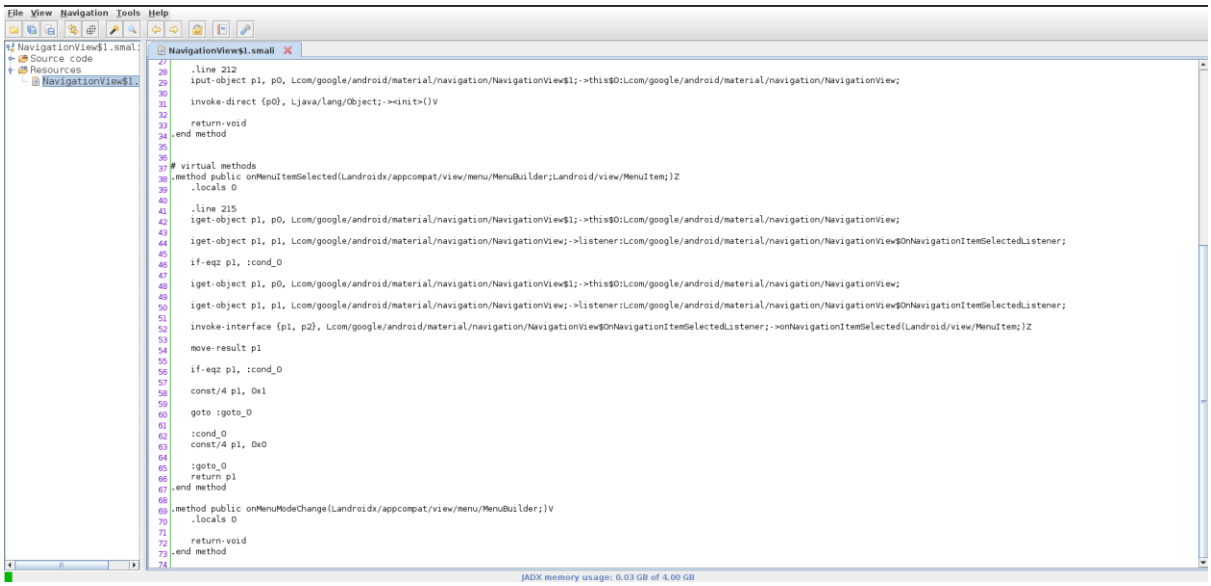
2. Reversing Process

Reversing the Application

The first thing to do is to reverse the application itself. I downloaded an .apk file from the PlayStore, Fietsknooppunt. With the apk, I used ApkTool to convert it into something readable.

```
kali@kali:/usr/local/bin$ sudo java -jar apktool.jar  d ~/Downloads/apk/Fietsknooppunt.apk
I: Using Apktool 2.5.0 on Fietsknooppunt.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package ...
I: Decoding file-resources ...
I: Decoding values */* XMLs ...
I: Baksmaling classes.dex ...
I: Baksmaling classes2.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...
kali@kali:/usr/local/bin$ ls
apktool  apktool.jar  evil-winrm  Fietsknooppunt  httpclient  nul  rwinrm  rwinrmcp  xsser
kali@kali:/usr/local/bin$
```

I turned it into Smali code, which is the human readable form of the language Android uses. It reads a bit like Assembly, but it actually tells you where something is going. It also shows you the source code in Java.



3. Findings

Interface + Base Activity

There is an enormous 'interface' class that has everything in it from every class. This may not be bad security wise, in a software aspect this is going against SOLID since 'a single class should only have a single responsibility.' They should have made separate interfaces for each class.

```
Source code
  nl.vv.fietsknoop
    Interface
      bearingLatLon(double, double, dou
      cleanupOpnemen(Context) void
      distanceLatLon(double, double, do
      getAfstand(Integer, boolean, int)
      getAfstanden(Boolean, int) String
      getColorBlue() int
      getColorGreen() int
      getColorGreenDark() int
      getColorOrange() int
      getColorRed() int
      getColorWhite() int
      getDateNl(String, int) String
      getDefaults(Context) String
      getFietsduur(Context, double, int
      getGPS(Context) String[]
      getHandleidingTekst(int) String[]
      getJsonStorage(Context, String) J
      getLand(int, int) String
      getLanguage(Context) int
      getLeeftijd(Integer, Boolean, int
      getLeeftijden(Boolean, int) Strin
      getLocation(Context) String[]
      getOffline(Context) String[]
      getOmgeving(Integer, int) String
      getOmgevingen(Boolean, int) Strin
      getPauze(Context) String[]
      getPosition(Context) String[]
      getPrivacy(int) String
      getProvincie(Integer, boolean, in
      getProvincies(Boolean, int) Strin
      getRegistration(Context) String[]
      getRegistreerAccount(Context) Str
      getScherm(Context) String[]
      getSexe(Integer, Boolean, int) St
      getSexen(Boolean, int) String[]
      getSnelheden() String[]
      getSnelheid(Integer) String
      getStringStorage(Context, String)
      getWaardering(Integer, int) Strin
      getWaarderingen(Boolean, int) Str
      getWachtwoordAccount(Context) Str
      md5(String) String
```

This keeps going.

```
import java.security.NoSuchAlgorithmException;
import org.json.JSONArray;
import org.json.JSONException;

public class Interface {
    public String[] getHandleidingTekst(int i) {
        int i2 = i;
        String[] strArr = new String[15];
        if (i2 == 1) {
            strArr[0] = "Make your own bike routes on the map. Click a bike
            strArr[1] = "Find your saved routes in this list. View the route
            strArr[2] = "See when you have cycled your routes. Have you recc
            strArr[3] = "Find a route, made by other Fietsknoop users. Set y
            strArr[4] = "Scan routes directly into the app from the Fietsknc
            strArr[5] = "National cycling routes (LF, RAVel and Véloroutes)
            strArr[6] = "Fietsknoop[+]Plus offers you the possibility to pla
            strArr[7] = "If you have Fietsknoop[+]Plus you can mark your fav
```

Neither is it an actual interface.

```
BaseActivity.smali
Source code
  p000nl.p001vv.fietsknoop
    BaseActivity
      RC_REQUEST int
      REQUEST_CODE_RECOVER_PLAY_SERVICES
      SKU_RECLAMEVRIJ_12_MAANDEN String
      TAG String
      baseAndroidId String
      baseRegistrationEm String
      baseRegistrationId Integer
      billingClient BillingClient
      count int
      delayModeGefietst double
      delayModeLFroutes double
      delayModeLabels double
      delayModeRoutes double
      delayModeVrienden double
      expireModeGefietst int
      expireModeLFroutes int
      expireModeLabels int
      expireModeRoutes int
      expireModeVrienden int
      inappBillingLog int
      inappPlusbedrag String
      languageSetting int
      lastTimeGefietst double
      lastTimeLabels double
      lastTimeRoutes double
      lastTimeVrienden double
      nextTimeLFroutes double
      packageVersionCode int
      packageVersionName String
      reclamevrijDetails SkuDetails
      startAppId String
      startDelayTimeSeconds long
      startRefreshTimeSeconds long
      baseLocationLocked Boolean
      {...} void
      alertErrorUnknown() void
      alertMessage(String) void
      alertMessageNoLocation() String
      alertNoAccount() void
      alertNoCamera() void
      alertNoData() void
      alertNoLink() void
```

There is also a huge file called ‘BaseActivity’ that certain classes extend. This is also going against Solid. It’s very long and not comprehensible.

POST – request

```java
public String postDataToUrl(String str, String str2) {
    String str3 = "";
    try {
        HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(str).openConnection();
        httpURLConnection.setReadTimeout(15000);
        httpURLConnection.setConnectTimeout(15000);
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setDoInput(true);
        httpURLConnection.setDoOutput(true);
        OutputStream outputStream = httpURLConnection.getOutputStream();
        BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
        bufferedWriter.write(str2);
        bufferedWriter.flush();
        bufferedWriter.close();
        outputStream.close();
        if (httpURLConnection.getResponseCode() == 200) {
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(httpURLConnection.getInputStream()));
            while (true) {
                String readLine = bufferedReader.readLine();
                if (readLine == null) {
                    break;
                }
                StringBuilder sb = new StringBuilder();
                sb.append(str3);
                sb.append(readLine);
                str3 = sb.toString();
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return str3;
}
```

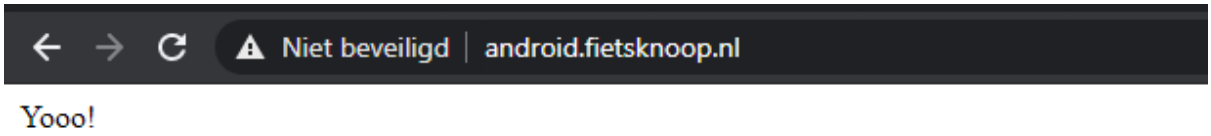
This is their code for a Post request, taken from the Interface file. It doesn't seem to have any checks if the endpoint should be reached. It just opens a connection, pushes it to POST, and then closes it again. It also takes the argument "str2", but it's not used by a lot of methods that do use this post request.

```java
    public String baseGetUrl(String str, String str2) {
        return baseGetUrlPrivate("android.fietsknoop.nl", str, str2);
    }

    public String baseGetWww(String str, String str2) {
        return baseGetUrlPrivate("www.fietsknoop.nl", str, str2);
    }
```

```java
private String baseGetUrlPrivate(String str, String str2, String str3) {
    int i;
    String str4;
    int i2 = 0;
    if (str3 == null || str3.length() == 0) {
        int intValue = baseRegistrationId.intValue();
        i = intValue;
        str3 = baseRegistrationEm;
    } else {
        i = 0;
    }
    if (VERSION.SDK_INT < 23) {
        i2 = 1;
        str4 = "http";
    } else {
        str4 = "https";
    }
    StringBuilder sb = new StringBuilder();
    sb.append(str4);
    sb.append("://");
    sb.append(str);
    sb.append("/mobile/40/index.php?nossl=");
    sb.append(i2);
    sb.append("&taal=");
    sb.append(languageSetting);
    sb.append("&mode=");
    sb.append(str2);
    sb.append("&device=android&deviceId=");
    sb.append(baseAndroidId);
    sb.append("&versionCode=");
    sb.append(packageVersionCode);
    sb.append("&version=");
    sb.append(packageVersionName);
    sb.append("&registId=");
    sb.append(i);
    sb.append("&email=");
    sb.append(str3);
    return sb.toString();
}
```

This is the BaseURL that other methods reference. When going to android.fietsknoop.nl, you get this:



It is quite dangerous to leave this as plaintext. There should be a better way to do this.

Password usage and storage

```java
public String[] getWachtwoordAccount(Context context) {
    String str = "";
    String[] strArr = {str};
    try {
        FileInputStream openFileInput = context.openFileInput("wachtwoord.fietsknoop");
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(openFileInput));
        while (true) {
            String readLine = bufferedReader.readLine();
            if (readLine == null) {
                break;
            }
            StringBuilder sb = new StringBuilder();
            sb.append(str);
            sb.append(readLine);
            str = sb.toString();
        }
        bufferedReader.close();
        openFileInput.close();
        String[] split = str.split(",");
        if (split.length > 0) {
            strArr[0] = split[0];
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return strArr;
}
```

This seems to imply that the password is written to a plaintext file within the application.

Conclusion

It seems secure enough, but very messy and a bit unprofessional. The user interface seems a bit outdated, and there are some cases where SOLID principles aren't followed. Further, the android.fietsknoop.nl url is not secured.