EZPZ APK

Programmi usati:
-unzip
-dex2jar
-JD-GUI
-android studio

Per l'analisi di questo apk, per iniziare, procedo alla decompressione (considerandolo come un comune formato .zip) utilizzando unzip:

```
unzip ezpz.apk
```

risulta in:

```
└─$ ls
AndroidManifest.xml  ezpz.apk                       firebase-database-collection.properties  play-services-basement.properties      res
classes2.dex         firebase-auth-interop.properties  firebase-firestore.properties            play-services-base.properties         resources.arsc
classes3.dex         firebase-common.properties        google                                   play-services-tasks.properties
classes.dex          firebase-components.properties    META-INF                                 protolite-well-known-types.properties
```

Notando la presenza di file denominati firebase,deduco vi sia la presenza di un eventuale database.

Per la prosecuzione dell'analisi, con lo strumento dex2jar (come si intuisce dal nome, converte i .dex in .jar) converto i 3 file classes(1,2,3).

```
└─$ d2j-dex2jar classes.dex classes2.dex classes3.dex
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
dex2jar classes.dex → ./classes-dex2jar.jar
dex2jar classes2.dex → ./classes2-dex2jar.jar
dex2jar classes3.dex → ./classes3-dex2jar.jar
```

E con lo strumento jd-gui, apro i file .jar per analizzarli:
(Per evitare passaggi inutili, passo direttamente a quello che interessa in questa sfida)
Quello che interessa trovare, in questo caso un main o funzione principale, si trova nel secondo file classes. Passando all'analisi del codice, evidenzio i seguenti punti importanti:

```java
public class MainActivity extends AppCompatActivity {
  EditText button;

  Button editText;

  int flag_counter = 0;

  protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131427356);
    this.editText = (Button)findViewById(2131231042);
    this.button = (EditText)findViewById(2131230881);
    if (!(new uselessClass()).flagCheckerxD((Activity)this))
      Toast.makeText(getApplicationContext(), "Ya need internet connection for the flag", 0).show();
    final String[] YEET = (new whyAmIHere()).isThisWhatUWant();
    this.editText.setOnClickListener(new View.OnClickListener() {
          public void onClick(View paramlView) {
            float[] arrayOfFloat = (new uselessClass()).toWhereEver(paramlView, context);
            if (MainActivity.this.button.getText().toString() != null) {
              if (MainActivity.this.flag_counter < 500) {
                MainActivity mainActivity = MainActivity.this;
                mainActivity.flag_counter++;
                MainActivity.this.editText.setX(arrayOfFloat[0]);
                MainActivity.this.editText.setY(arrayOfFloat[1]);
                Toast.makeText(MainActivity.this.getApplicationContext(), "Lets Play :)", 0).show();
              } else if (YEET[0].equals(MainActivity.this.button.getText().toString())) {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Well thats the  Correct Flag", 0).show();
              } else {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Damn...500 times? are u kidding me", 0).show();
              }
            } else {
              Toast.makeText((Context)context, "Gib flag or get out", 0).show();
            }
          }
        });
  }
}
```

Si nota come:
-La flag risiede all'interno della variabile "**YEET**", all'indice 0. Il valore di "**YEET**" viene dichiarato dall'istanziamento di un oggetto "**whyAmIHere()**", con il metodo "**isThisWhatUWant()**"

-l'effettiva verifica della stringa inserita con la flag memorizzata avviene dopo 499 inserimenti

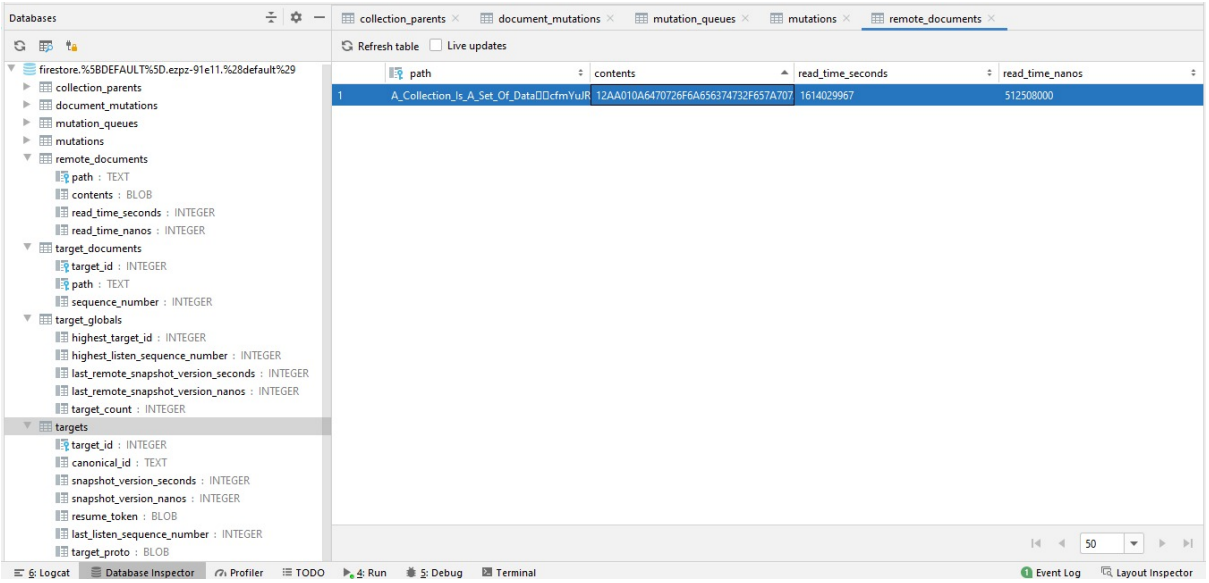
procedo dunque all'analisi della classe "**isThisWhatUWant()**":

```java
public class whyAmIHere {
  public String[] isThisWhatUWant() {
    final String[] justAWaytoMakeAsynctoSync = new String[1];
    arrayOfString[0] = "";
    FirebaseFirestore.getInstance().collection("A_Collection_Is_A_Set_Of_Data").get().addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
          public void onSuccess(QuerySnapshot paramlQuerySnapshot) {
            for (DocumentSnapshot documentSnapshot : paramlQuerySnapshot) {
              justAWaytoMakeAsynctoSync[0] = documentSnapshot.getString("Points");
              Log.d("TypicalLogcat", justAWaytoMakeAsynctoSync[0]);
            }
          }
        }).addOnFailureListener(new OnFailureListener() {
          public void onFailure(Exception paramlException) {
            justAWaytoMakeAsynctoSync[0] = "Something Failed,Maybe Contact Author?";
          }
        });
    return arrayOfString;
  }
}
```

si nota come:
-venga dichiarato un array di stringhe vuoto, nessun metodo aggiunge valori, e viene fatto il return del valore vuoto.
-Viene fatta una richiesta per ottenere dei dati, conservata in una collezione di documenti chiamata "**A_Collection_Is_A_Set_Of_Data**".
-I valori ottenuti dal successo della richiesta, non vengono utilizzati, solo conservati.

Dunque, per trovare la flag, devo analizzare il traffico dati dell'applicazione. Per farlo, utilizzo Android Studio, in modo da tenere traccia del database interno:

Ricapitolando prima della ricerca: Devo trovare una collezione di documenti sotto la denominazione “**A_Collection_Is_A_Set_Of_Data**”.



Quello che interessa nell’analisi, riguarda i documenti. In questo caso, quello che in cui si riscontrano le caratteristiche che cerchiamo è in **“remote_documents”**.

Dunque copio il valore nella colonna del contenuto:
“*12AA010A6470726F6A656374732F657A707A2D39316531312F6461746162617365732F2864656661756C74292F646F63756D656E74732F415F436F6C6C656374696F6E5F49735F415F5365745F4F665F446174612F63666D59754A52797053426A69614F72766C6373123400A06506F696E7473122A8A01276461726B434F4E7B64336275675F6D35675F316E5F7072306475637431306E5F31735F6234647D220C088FD3BD810610809AF0B601*”
Riconosco sia una codifica in esadecimale, dunque decodificando risulta in:
a

dprojects/ezpz-91e11/databases/(default)/documents/A_Collection_Is_A_Set_Of_Data/cfmYuJRypSBjiaOrvlcs4
Points*Š'**darkCON{d3bug_m5g_1n_pr0duct10n_1s_b4d}**" Ó½ šð¶

-GreyHat