CÁTEDRA DE CIBERSEGURIDAD CIBERUGR, INCIBE-UGR

Nombre	GoodNote
Categoría	REVERSING
Dificultad	MEDIA
Puntos	300

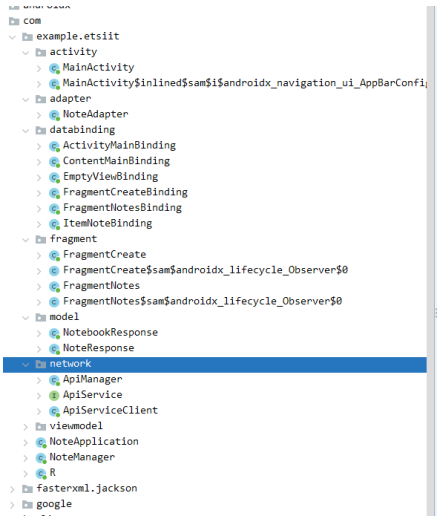
DESCRIPCIÓN DEL RETO

No me fio de las aplicaciones de notas de la PlayStore. ¡¡¡NOS ESTAN ESPIANDO!!! Por ello, he aquí mi aplicación de notas, donde aseguro que tus notas serán 1000% confidenciales.

Firmado, tu programador de confianza.

WRITEUP

1. Abrir el APK con jadx-gui (https://github.com/skylot/jadx). Podemos observar que, para obtener las notas, la aplicación se conecta a un servidor remoto.



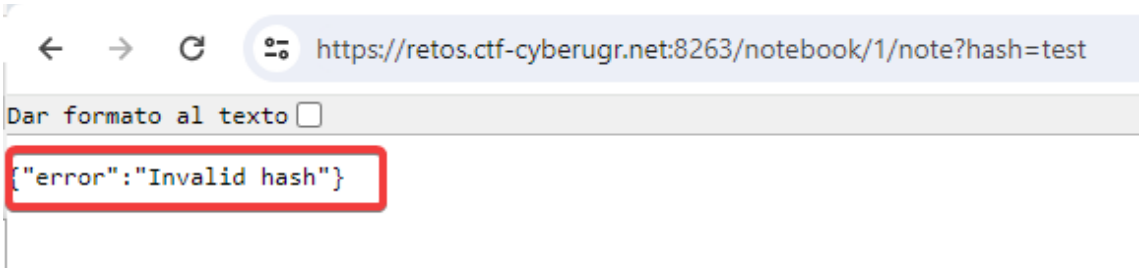
2. Dentro de la carpeta “network”, en la clase “ApiServiceClient”, observar el dominio del *endpoint* a donde se conecta.

```
public final class ApiServiceClient {
    private static final String BASE_URL = "https://retos.ctf-cyberugr.es:8263/";
    public static final ApiServiceClient INSTANCE = new ApiServiceClient();
    private static final Lazy retrofit$delegate = LazyKt.lazy(new Function0<Retrofit>() { // from class: com.example.etsiit.network.ApiServiceClient$retrofit$2
        @Override // kotlin.jvm.functions.Function0
        public final Retrofit invoke() {
            return new Retrofit.Builder().baseUrl("https://retos.ctf-cyberugr.es:8263/").addConverterFactory(GsonConverterFactory.create()).build();
        }
    });
    private static final Lazy apiService$delegate = LazyKt.lazy(new Function0<ApiService>() { // from class: com.example.etsiit.network.ApiServiceClient$apiService$2
        /* JADX WARN: Can't rename method to resolve collision */
        @Override // kotlin.jvm.functions.Function0
        public final ApiService invoke() {
            Retrofit retrofit;
            retrofit = ApiServiceClient.INSTANCE.getRetrofit();
            return (ApiService) retrofit.create(ApiService.class);
        }
    });
```

3. Dentro de la carpeta “network”, en la clase “ApiService”, observar los endpoints de la API a donde se conecta

```
/* compiled from: ApiService.kt */
@Metadata(d1 = {"\u00006\n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0000\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0000\n\u0002\u
/* loaded from: classes.dex */
public interface ApiService {
    @FormUrlEncoded
    @POST("/notebook/{notebookId}/note/create")
    Call<NoteResponse> createNote(@Path("notebookId") int i, @Field("name") String str, @Field("content") String str2);

    @POST("/notebook/create")
    Call<NotebookResponse> createNotebook();

    @GET("/notebook/instance")
    Call<InstanceResponse> getInstance();

    @GET("/notebook/{notebookId}/note/{noteId}")
    Call<NoteResponse> getNote(@Path("notebookId") int i, @Path("noteId") int i2, @Query("hash") String str);

    @GET("/notebook/{notebookId}")
    Call<NotebookResponse> getNotebook(@Path("notebookId") int i);

    @GET("/notebook/{notebookId}/note")
    Call<List<NoteResponse>> getNotes(@Path("notebookId") int i, @Query("hash") String str);
}
```

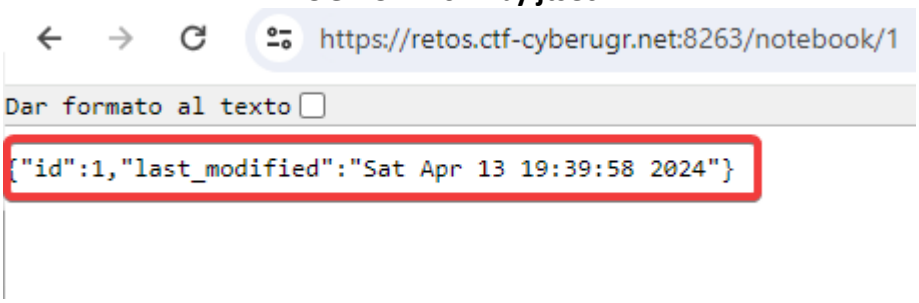
4. Sabiendo los *endpoints* a los que se conecta, podemos usar el mismo navegador para intentar obtener una nota. Para ello, haremos una petición al libro número 1 para ver si somos capaces de sacar la nota con el hash “test”.

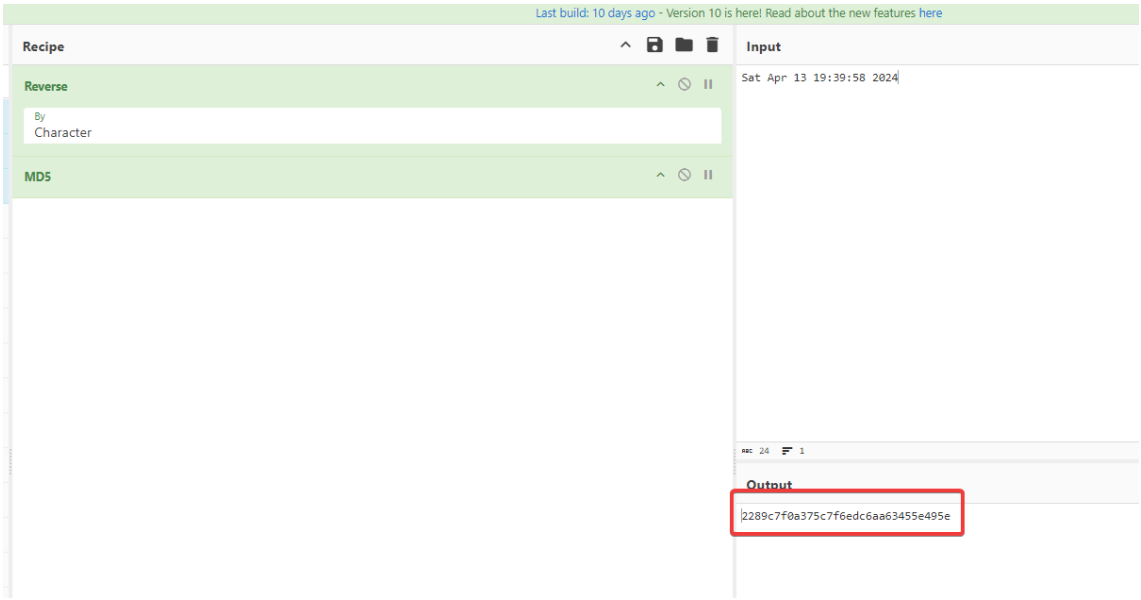


5. Analizamos la función “invoke2”, la cual obtiene la fecha que devuelve la libreta, le da la vuelta y realiza un hash md5 a todo ello

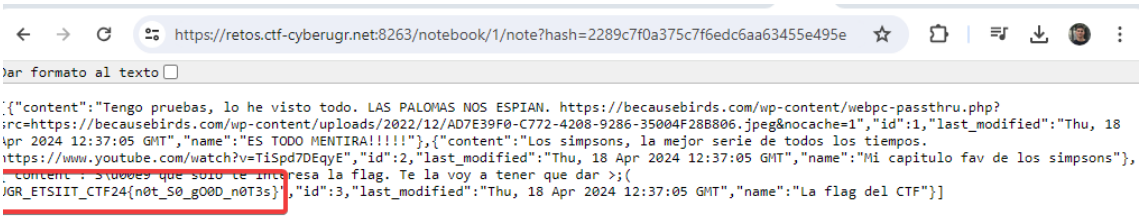
```
        /* renamed from: invoke  reason: avoid collision after fix types in other method */
        public final void invoke2(NotebookResponse notebookResponse) {
            NotebookViewModel notebookViewModel;
            NoteManager noteManager2;
            if (notebookResponse != null) {
                final FragmentNotes fragmentNotes = FragmentNotes.this;
                int i2 = i;
                MessageDigest messageDigest = MessageDigest.getInstance("MD5");
                byte[] bytes = StringsKt.reversed((CharSequence) notebookResponse.getNotebookLastModified()).toString().getBytes(Charsets.UTF_8);
                Intrinsics.checkNotNullExpressionValue(bytes, "getBytes(...)");
                String bigInteger = new BigInteger(1, messageDigest.digest(bytes)).toString(16);
                Intrinsics.checkNotNullExpressionValue(bigInteger, "toString(...)");
                String padStart = StringsKt.padStart(bigInteger, 32, '0');
                notebookViewModel = fragmentNotes.getNotebookViewModel();
                notebookViewModel.setNotebook(notebookResponse);
```

6. Para obtener el hash, primero obtenemos la fecha de modificación y en CyberChef (https://gchq.github.io/CyberChef/) realizamos las operaciones antes descritas.





7. Para obtener las notas, usaremos el hash resultante en el parámetro hash de la petición GET de la libreta 1.



8. Finalmente, en la petición anterior, consiguiendo acceder a las notas del libro 1, podemos observar como un usuario ha dejado la *flag*.