vishwaCTF

CHALLENGE NAME: [PHANTON ROLLCALL]

DEV : [JAY RAJANKAR]

CATEGORY: [REVERSE ENGINEERING]

LEVEL: [MEDIUM]

2025

Challenge Description:

The system only acknowledges those who follow the unseen path. Attendance must be marked, but the method remains obscured. A hidden truth lingers beneath the surface—buried where only the keen-eyed dare to look.

Trace the echoes, uncover the secret, and claim your presence.

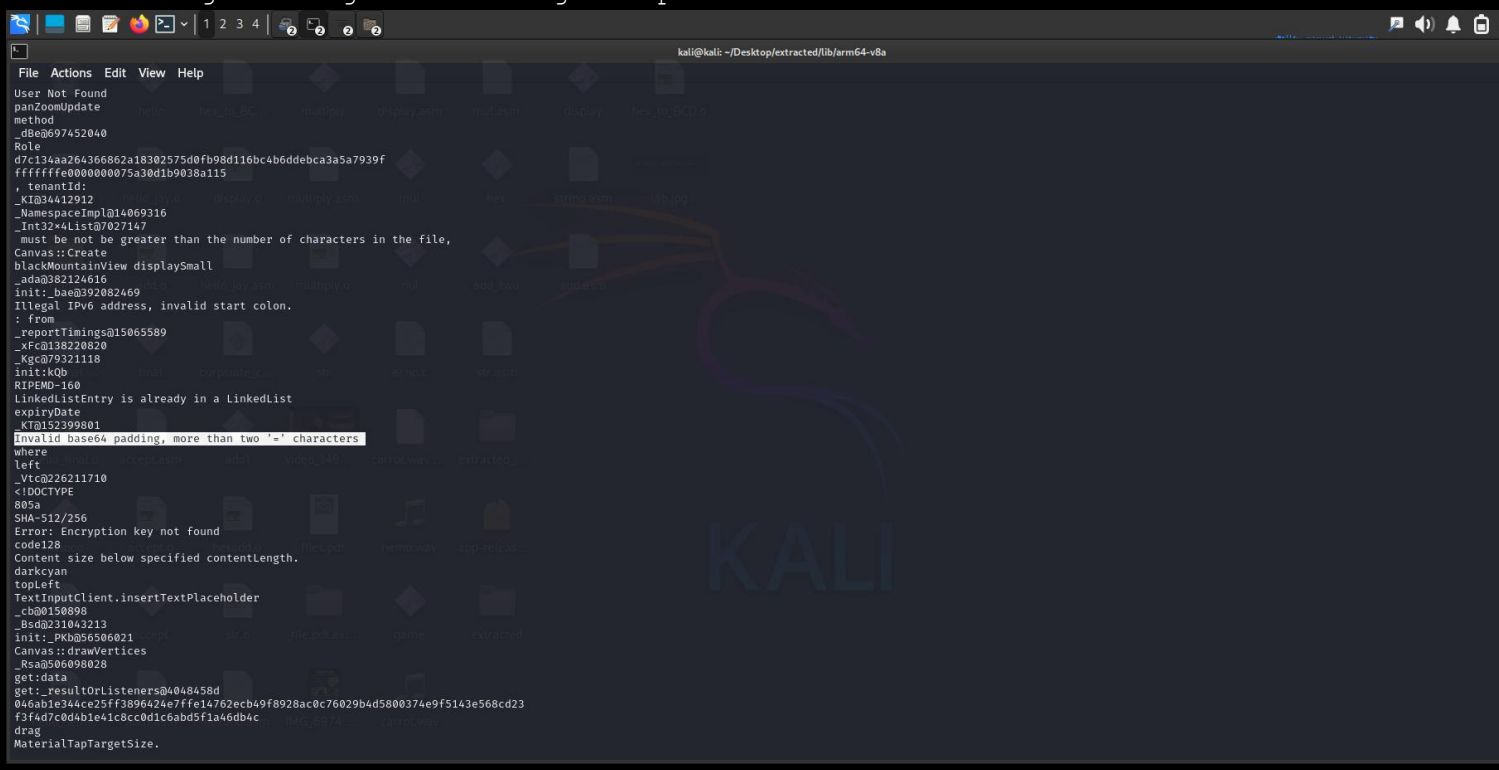
Solution:

We are given a file app-release.apk , As the question is in reverse engineering domain we should analyse how the app works, the description says to trace the echoes , which means we have to trace something and uncover secret , when opened the app it feels like a attendance app which marks attendance based on qr or entered string.

The last line of the question can be interpreted as we have to mark the attendance to clear the final stage.

Now into the reverse engineering part , after decompiling the APK we can clearly guess that it is a flutter app .Hence the logic and code written by user is always in libapp.so . Reverse engineering obfuscated code can be approached in multiple ways, but whether applying logic or analyzing strings is more effective depends on the level and type of obfuscation used. Firstly, we will try extracting strings from the libapp.so should be the easiest way to read meaningful strings a obfuscated code.

We have to go through the strings outputted and see what makes sense.



Here we can see a check for whether the string is a base 64 or not

```
kali@kali: ~/Desktop/extracted/lib/arm64-v8a
File  Actions  Edit  View  Help
primary_fixed
_Sra@258328197
handleValue
_hUa@1026248
d35e472036bc4fb7e13c785ed201e065f98fcfa6f6f40def4f92b9ec7893ec28fcd412b1f1b32
e24
_OS.
QrValidationStatus.
ImageFilter::initColorFilter>
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.regis
terIdTokenListener
_vYb@205384654
_ydc@143441002
TVInputComponent1
_qa@0150898
Cannot clear an unmodifiable list
_hA@321411118
_Uda@143441002
_Zra@259099430
PlatformConfigurationNativeApi::SendPlatformMessage$
A more efficient solution is to split your build function into several widget
s. This introduces a new context from which you can obtain the Scaffold. In t
his solution, you would have an outer widget that creates the Scaffold popula
ted by instances of your new inner widgets, and then in these inner widgets y
ou would use Scaffold.of().
A less elegant but more expedient solution is assign a GlobalKey to the Scaff
old, then use the key.currentState property to obtain the ScaffoldState rathe
r than using the Scaffold.of() function.
TransitionRoute
_Uint16List@7027147
"(?:[^^\x00-\x1F\x7F]|\\.)*"
dart:math
_bc@4048458
Hiragana
 p -
proxy-authorization
rosybrown
plum
TextInput.setMarkedTextRect
RegExp_getGroupCount
_iVc@168042876
_vQ@108420462
named
QrInputTooLongException:
clip-path
^SHA-512\/([0-9]+)$
' has no instance
Time:
TVAudioDescriptionMixDown
FavoriteClear1
```

Here we can see that the app is using firebase authentication, which is used for user authentication

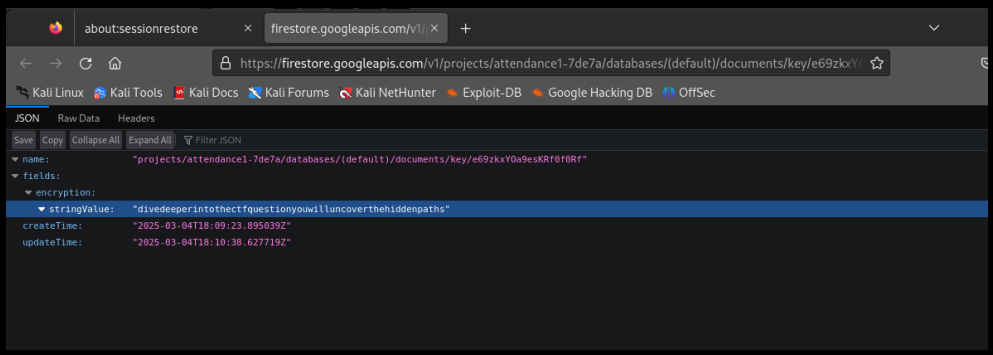
```
kali@kali: ~/Desktop/extracted/lib/arm64-v8a
File  Actions  Edit  View  Help
_qGe@802155622
title
_un@13463476
_fY@296160605
init:_HUd@366461389
, patternDataId:
TVNumberEntry
_jab@15065589
_yme@498312174
plugins.flutter.io/firebase_firestore
scopesRoute
_UA@325511922
_bg@9040228
forestgreen
Cannot extract a non-Windows file path from a file URI with an authority
mediumvioletred
["*/:<>?\\]
_qN@74331726
isChecked
_AsyncStreamController@4048458
init:_Hke@481464419
Internal_makeListFixedLength
7ffffffffffffffffffffff800000cfa7e8594377d414c03821bc582063
_uWc@169492240
signed
/PKCS12
PROXY
_eIa@0150898
set:_errorCode@14069316
moveRight:
primary_fixed
_Sra@258328197
handleValue
_hUa@1026248
d35e472036bc4fb7e13c785ed201e065f98fcfa6f6f40def4f92b9ec7893ec28fcd412b1f1b32
e24
_OS.
QrValidationStatus.
ImageFilter::initColorFilter>
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.regis
terIdTokenListener
_vYb@205384654
_ydc@143441002
TVInputComponent1
_qa@0150898
Cannot clear an unmodifiable list
_hA@321411118
_Uda@143441002
_Zra@259099430
PlatformConfigurationNativeApi::SendPlatformMessage$
```

here we can see that the app is also using another service by firebase , i.e. firebase firestore , which is a database service. Going back to question , it tells us to trace. Lets check where the firestore is used .

```
┌──(kali㉿kali)-[~/Desktop/extracted/lib/arm64-v8a]
└─$ strings libapp.so | grep "firebase"
firebase_auth
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.signInWithEmailAndPassword
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.signOut
plugins.flutter.io/firebase_firestore/query/
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.createUserWithEmailAndPassword
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.registerAuthStateListener
plugins.flutter.io/firebase_firestore
dev.flutter.pigeon.firebase_auth_platform_interface.FirebaseAuthHostApi.registerIdTokenListener
plugins.flutter.io/firebase_auth

┌──(kali㉿kali)-[~/Desktop/extracted/lib/arm64-v8a]
└─$ strings libapp.so | grep "firestore"
https://firestore.googleapis.com/v1/projects/attendance1-7de7a/databases/(default)/documents/key/e69zkxYOa9esKRf0f0Rf
firestore
dev.flutter.pigeon.cloud_firestore_platform_interface.FirebaseFirestoreHostApi.documentReferenceGet
dev.flutter.pigeon.cloud_firestore_platform_interface.FirebaseFirestoreHostApi.documentReferenceSet
plugins.flutter.io/firebase_firestore/query/
dev.flutter.pigeon.cloud_firestore_platform_interface.FirebaseFirestoreHostApi.queryGet
dev.flutter.pigeon.cloud_firestore_platform_interface.FirebaseFirestoreHostApi.querySnapshot
plugins.flutter.io/firebase_firestore
cloud_firestore
```

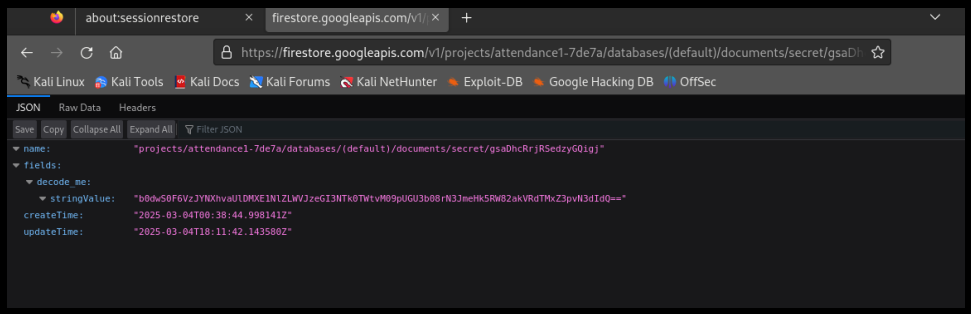
We can clearly see that a database is being accessed via a hardcoded link. Let's see contents of the collection key.



So this seems to be a encryption key or something or more like a hint. (the contain of this may vary as during the event the string was updated)
As we have to search for all of the paths that the app accesses of the database we will simply search for the thing that itself defines a path "/".
So when executed strings libapp.so | grep "/"

```
kali@kali: ~/Desktop/extracted/lib/arm64-v8a
File  Actions  Edit  View  Help
bi~/
/v4b
$m//
 `h/
./]J
/5k.
/4j,
Invalid arguments for 'resize' method sent to dev.flutter/channel-buffers (channel name must be less than 254 characters long)
text/plain
Invalid arguments for 'overflow' method sent to dev.flutter/channel-buffers (second argument must be a boolean)
/CCM
flutter/textinput
/secret/gsaDhcRrjRSedzyGQigj
flutter/lifecycle
(^[a-zA-Z][-+.a-zA-Z\d]*://|[^/])$
EEE, M/d/y
dev.steenbakker.mobile_scanner/scanner/method
flutter/keydata
flutter/mousecursor
https://firestore.googleapis.com/v1/projects/attendance1-7de7a/databases/(default)/documents/key/e69zkxYOa9esKRf0f0Rf
^(.+)/([^/]+)$
^(.+)/OFB-([0-9]+)$
Invalid arguments for 'overflow' method sent to dev.flutter/channel-buffers (first argument must be a string)
^(.*)/CTR/AUTO-SEED-PRNG$
flutter/keyevent
assets/images/main_top.png
flutter/processtext
/ConcatKDF
^(.+)/(DET-)?ECDSA$
flutter/system
file:///
Key length not 128/192/256 bits.
Unrecognized message sent to dev.flutter/channel-buffers (method name too long)
Dart/
^(\\\[^\\]+\\[^\\/]+|[a-zA-Z]:[/\\])"
[\\]
package:flutter/src/services/message_codec.dart
 !"#$%&'()*+,-./0123456789:;<=>?@abcdefghijklmnopqrstuvwxyz[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
M/d/y
2/?O=O=O=O=O=O=O=
plugins.flutter.io/firebase_firestore/query/
Keccak/
/CTR
Invalid arguments for 'resize' method sent to dev.flutter/channel-buffers (first argument must be a string)
/EAX
/PKCS1
/PKCS5S1
Invalid arguments for 'overflow' method sent to dev.flutter/channel-buffers (channel name must be less than 254 characters long)
/HMAC
^(.+)/CFB-([0-9]+)$
```

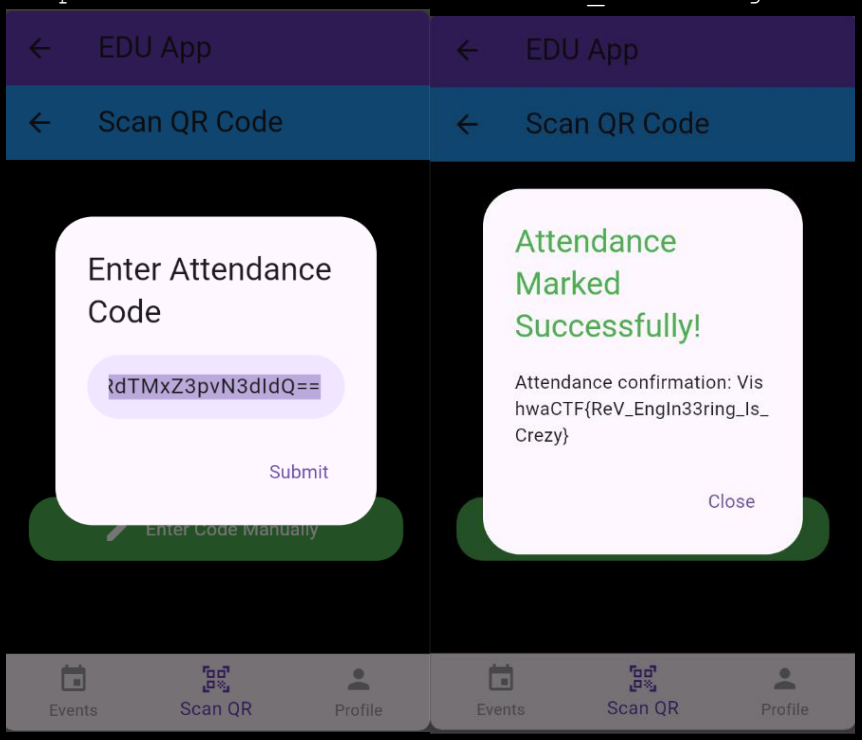
we can see that there is a path of a collection that looks similar to collection and document path of the collection key, also the question also mentions uncover the secret. The question is we don't have the link or entire path of the co llection . so we try to replace "/key/e69zkxYOa9esKRf0f0Rf " from "https://firestore.googleapis.com/v1/projects/attendance1-7de7a/databases/(default)/documents/key/e69zkxYOa9esKRf0f0Rf" with "secret/gsaDhcRrjRSedzyGQigj". So we get "https://firestore.googleapis.com/v1/projects/attendance1-7de7a/databases/(default)/documents/secret/gsaDhcRrjRSedzyGQigj"



Decode me seems to be in base 64 . lets recollect the app checks for base 64 and accesses the encryption key , that means it should have a decoder and decrypter present in the app .

Now there are two ways to proceed

simple one is to enter the decode_me string in the app attendance marking system



the second method is finding out which encryption method is actually being used .Firstly decoding the base 64 string, For finding the encryption used we will search for the encryption methods in strings.

and we have got the encryption key from the database

decrypting will reveal the flag.

Flag: VishwaCTF{ReV_EngIn33ring_Is_Crezy}