ANALYSE DU GESTIONNAIRES DE MOTS DE PASSE

✓ **REALISEATION :**

- **EL BEKHTI AYOUB**
- **HAIMOUD YASSINE**
- **ISMAIL ARAME**

✓ **ENCADRÉ PAR:**

- **Mr.KOULALI MOHAMMED AMINE**

SICS 4 **2023/2024**

❖ Presentation de projet et leur but:

Dans ce monde de plus en plus connecté, où les données personnelles sont devenues une monnaie d'échange virtuelle, la sécurité des informations sensibles est cruciale. Les cybermenaces évoluent constamment, des attaques sophistiquées de phishing aux vulnérabilités zero-day, menaçant la confidentialité et l'intégrité des données.

Les applications de gestionnaires de mots de passe se positionnent comme une première ligne de défense essentielle dans la protection des informations personnelles. En consolidant l'accès derrière un seul point d'entrée sécurisé, ces applications réduisent le risque d'exposition due à des pratiques de gestion de mots de passe inadéquates, telles que l'utilisation de mots de passe faibles ou la réutilisation de mots de passe sur plusieurs plateformes.

Dans ce projet on souhaite de faire une analyse dynamique et statique de certaines gestionnaires de mots de passe et tester leur security , mais avant faire ca c'est quoi un gestionnaire de mots passe ?

Un gestionnaire de mots de passe est une application conçue pour aider les utilisateurs à gérer et à stocker de manière sécurisée leurs informations d'identification en ligne, telles que les noms d'utilisateur et les mots de passe. Ces outils sont particulièrement utiles à mesure que les utilisateurs accumulent un grand nombre de comptes sur différents services en ligne.

➢ POUR QUOI ON UTILISE LES GESTIONNAIRES DE MOT S DE PASSE ?

Les gestionnaires de mots de passe sont utilisés pour plusieurs raisons importantes, visant à améliorer la sécurité et la gestion des informations d'identification en ligne. Voici quelques-unes des principales raisons pour lesquelles les gens utilisent des gestionnaires de mots de passe :

- **Complexité des Mots de Passe : Les recommandations en matière de sécurité suggèrent l'utilisation de mots de passe complexes, difficiles à deviner. Cependant, la gestion de nombreux mots de passe complexes peut être difficile sans assistance. Les gestionnaires de mots de passe permettent de générer, stocker et gérer automatiquement des mots de passe forts.**

- **Sécurité Renforcée : Les gestionnaires de mots de passe utilisent souvent des techniques de cryptage avancées pour sécuriser les informations d'identification. Cela ajoute une couche de protection supplémentaire par rapport à la mémorisation manuelle des mots de passe ou à la réutilisation de mots de passe sur plusieurs comptes.**

- **Facilité d'Utilisation : Les gestionnaires de mots de passe simplifient le processus d'authentification en ligne. Les utilisateurs n'ont pas besoin de se souvenir de plusieurs mots de passe, car le gestionnaire peut remplir automatiquement les champs d'identification.**

- **Gestion de Nombreux Comptes : À mesure que les utilisateurs s'inscrivent à de plus en plus de services en ligne, la gestion des informations d'identification devient une tâche complexe.**

❖ Les outils qu'on va utilisé :

- **JADX :**

est un outil open-source utilisé pour décompiler des fichiers DEX (Dalvik Executable) issus d'applications Android (.apk) en fichiers Java source. Les fichiers DEX sont le résultat de la compilation des fichiers sources Java vers le format exécutable pour la machine virtuelle Dalvik, qui était utilisée par Android avant l'adoption de la machine virtuelle ART (Android Runtime).

- **BURPSUITE :**

Burp Suite est une suite d'outils de sécurité des applications web développée par PortSwigger. C'est un ensemble d'outils utilisés par les professionnels de la sécurité informatique pour effectuer des tests d'intrusion et des analyses de sécurité sur les applications web.

- **FRIDA :**

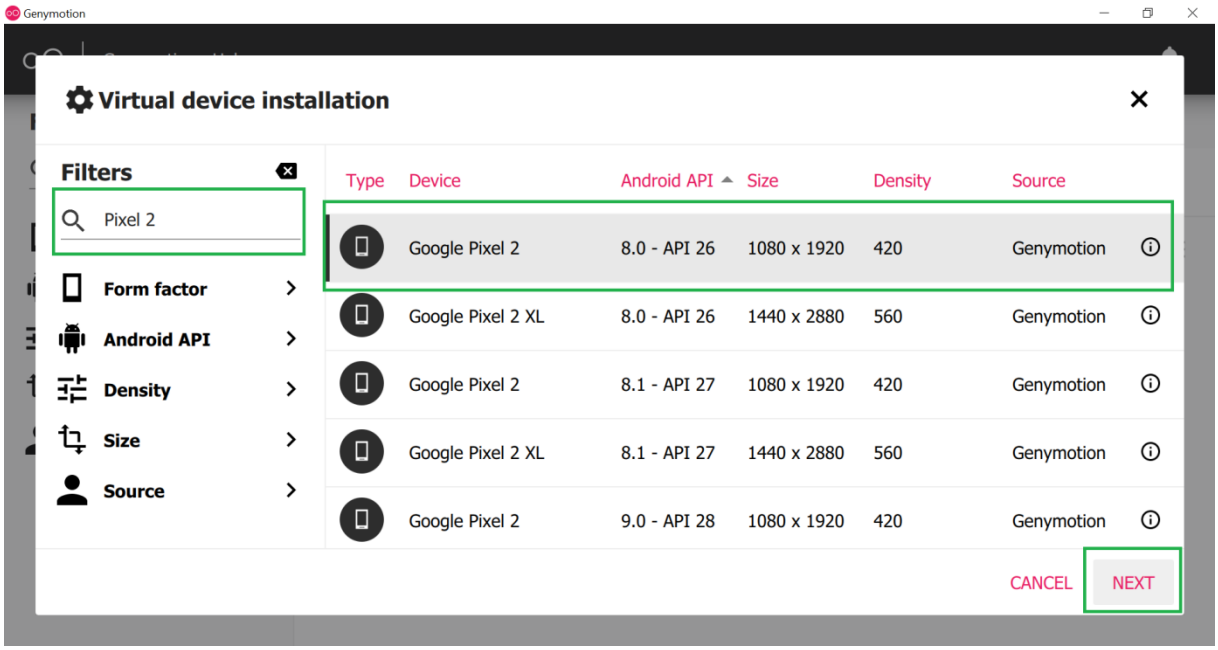
Frida est un framework open-source permettant le dynamic instrumentation d'applications, ce qui signifie qu'il offre la possibilité de manipuler et de modifier le comportement d'applications en cours d'exécution sur divers systèmes d'exploitation. Il est souvent utilisé dans le domaine de la sécurité informatique pour effectuer des tests d'intrusion, des analyses de sécurité, et pour la recherche en sécurité des applications.

- **OBJECTION :**

est un runtime mobile en temps réel, alimentée par Frida, conçue pour vous aider à évaluer la posture de sécurité de vos applications mobiles sans avoir besoin d'un jailbreak.

- **GENYMOTION :**

Genymotion est un émulateur Android puissant et polyvalent conçu pour simplifier le processus de développement et de test d'applications Android. Il offre un environnement virtuel permettant d'exécuter des applications Android sur différents dispositifs virtuels.



- **ADB :**

ADB (Android Debug Bridge) est une interface de ligne de commande qui fait partie du kit de développement Android (Android SDK). Elle permet une communication entre un ordinateur et un appareil Android, que ce soit un smartphone, une tablette, ou un autre dispositif.

❖ Les applications android qu'on va analyser :

Keeper : Keeper provides useful extra features like an encrypted messaging service and has more customer support options.

NordPass : NordPass includes more cloud storage and uses a more "futureproof" encryption.

mSecure: basic password manager features like password sharing, two-factor authentication, a password generator, autofill, and a password strength report to help keep hackers at bay.

Bitwarden :is best for users looking for a platform with advanced security measures and insights and the most affordable pricing.

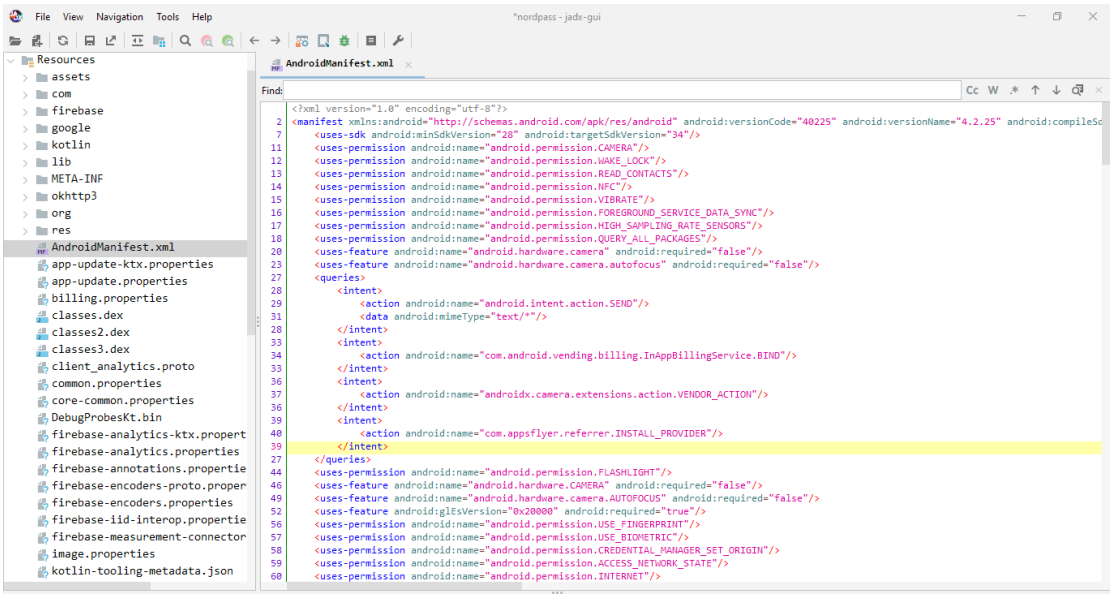
bitwarden

❖ Partie technique : analyse statique et dynamique des applications :

➢ NordPass :

Static Analysis :

✓ On démarre l'exploration du fichier NordPass.apk dans Jadx, et comme d'habitude, on entame notre analyse en examinant les autorisations dans le fichier AndroidManifest.xml :



✓ NordPass demande à chaque user de créer son propre master password qui est la clé de son coffre-fort. Grâce à lui, il autorise NordPass à réaliser sa fonction : stocker, récupérer et remplir automatiquement les mots de passe à sa place :

```
package com.nordpass.android.ui.authentication.masterpassword.create;

CreatePasswordConfirmFragment createPasswordConfirmFragment = (CreatePasswordConfirmFragment) this.f26281s;
int i12 = CreatePasswordConfirmFragment.I;
uc0.l.f(createPasswordConfirmFragment, "this$0");
CreatePasswordConfirmViewModel h11 = createPasswordConfirmFragment.h();
String str = ((go.a) createPasswordConfirmFragment.H.getValue()).f13766a;
String text = ((i3) createPasswordConfirmFragment.N()).R.getText();
h11.getClass();
uc0.l.f(str, "masterPassword");
uc0.l.f(text, "confirmMasterPassword");
h11.m(true, new go.d(h11, str, text, null));
return;
```

- ✓ Pour le chiffrement & déchiffrement, NordPass utilise l'algorithme XChaCha20 qui est un moyen de chiffrer et de déchiffrer les données. Il prend en charge deux longueurs de clés différentes ; le chiffrement 256 bits étant le plus fort. NordPass utilise XChaCha20 pour chiffrer notre coffre-fort de mots de passe :

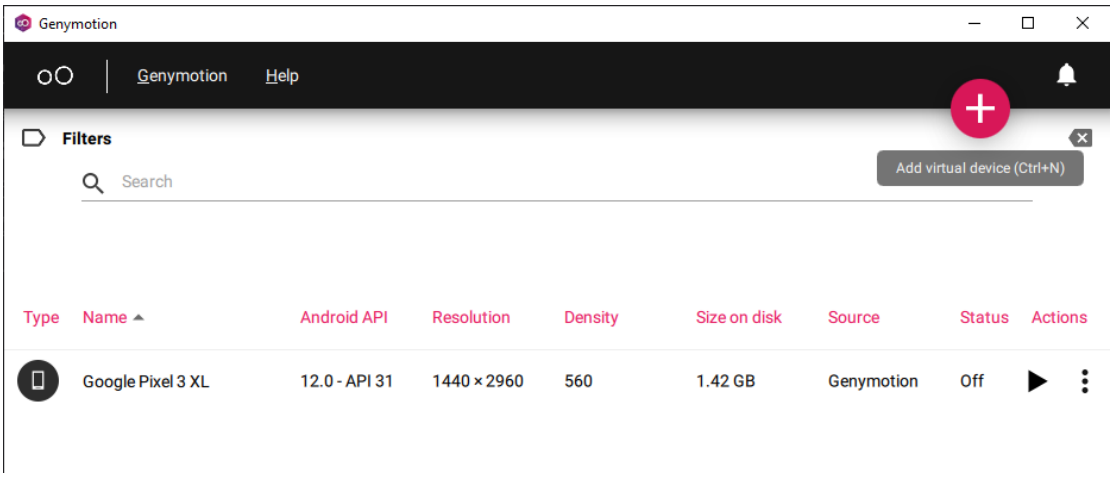
```
43    static {
44        HashMap hashMap = new HashMap();
45        hashMap.put(Ciphers.TLS_AES_128_GCM_SHA256, "AEAD-AES128-GCM-SHA256");
46        hashMap.put(Ciphers.TLS_AES_256_GCM_SHA384, "AEAD-AES256-GCM-SHA384");
47        hashMap.put(Ciphers.TLS_CHACHA20_POLY1305_SHA256, "AEAD-CHACHA20-POLY1305-SHA256");
48        j2oTls13 = Collections.unmodifiableMap(hashMap);
49        HashMap hashMap2 = new HashMap();
50        hashMap2.put(Ciphers.TLS_AES_128_GCM_SHA256, Collections.singletonMap("TLS", Ciphers.TLS_AES_128_GCM_SHA256));
51        hashMap2.put(Ciphers.TLS_AES_256_GCM_SHA384, Collections.singletonMap("TLS", Ciphers.TLS_AES_256_GCM_SHA384));
52        hashMap2.put(Ciphers.TLS_CHACHA20_POLY1305_SHA256, Collections.singletonMap("TLS", Ciphers.TLS_CHACHA20_POLY1305_SHA256));
53        hashMap2.put("AEAD-AES128-GCM-SHA256", Collections.singletonMap("TLS", Ciphers.TLS_AES_128_GCM_SHA256));
54        hashMap2.put("AEAD-AES256-GCM-SHA384", Collections.singletonMap("TLS", Ciphers.TLS_AES_256_GCM_SHA384));
55        hashMap2.put("AEAD-CHACHA20-POLY1305-SHA256", Collections.singletonMap("TLS", Ciphers.TLS_CHACHA20_POLY1305_SHA256));
56        o2jTls13 = Collections.unmodifiableMap(hashMap2);
57    }
```

- ✓ On voit l'usage de ChaCha20-Poly1305, c'est un algorithme de chiffrement authentifié avec données supplémentaires (AEAD), qui combine le chiffrement de flux ChaCha20 avec le code d'authentification de message Poly1305. Ce dernier peut être utilisé pour authentifier un seul message à l'aide d'une clé partagée entre le client et le NordPass server. - Pour la génération et le stockage des clés , ce password manager utilise AndroidKeyStore comme un conteneur des clés qui sont générées à partir de l'algorithme AES en mode CBC:

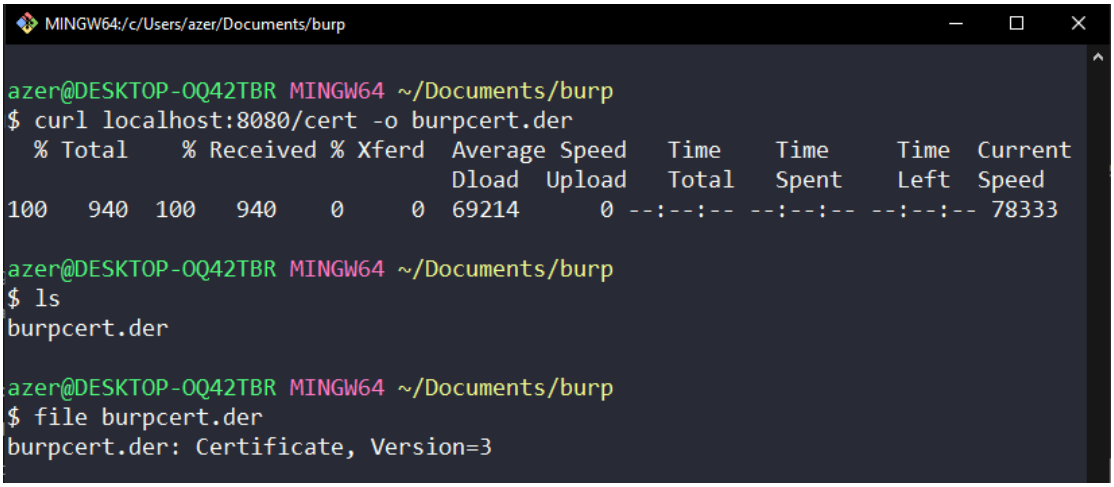
```
KeyStore keyStore = KeyStore.getInstance("AndroidKeyStore");
keyStore.load(null);
KeyGenParameterSpec.Builder b12 = y.a.b("androidxBiometric", 3);
y.a.d(b12);
y.a.e(b12);
KeyGenerator keyGenerator = KeyGenerator.getInstance("AES", "AndroidKeyStore");
y.a.c(keyGenerator, y.a.a(b12));
keyGenerator.generateKey();
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS7Padding");
cipher.init(1, (SecretKey) keyStore.getKey("androidxBiometric", null));
cVar = new BiometricPrompt.c(cipher);
```

Dynamic Analysis :

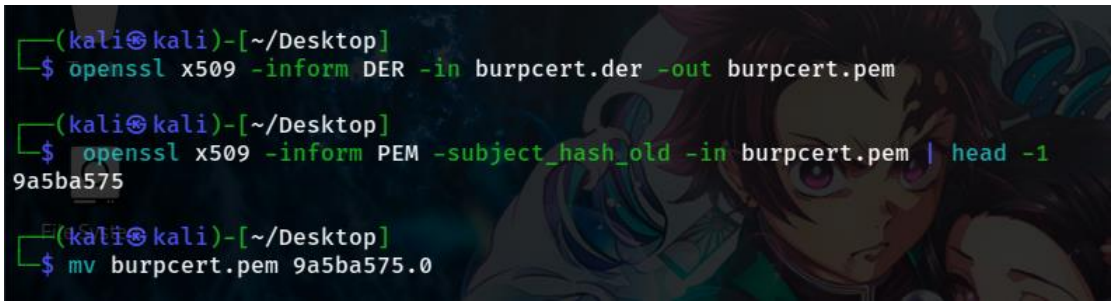
✓ Confuguring burp suite with genymotion :



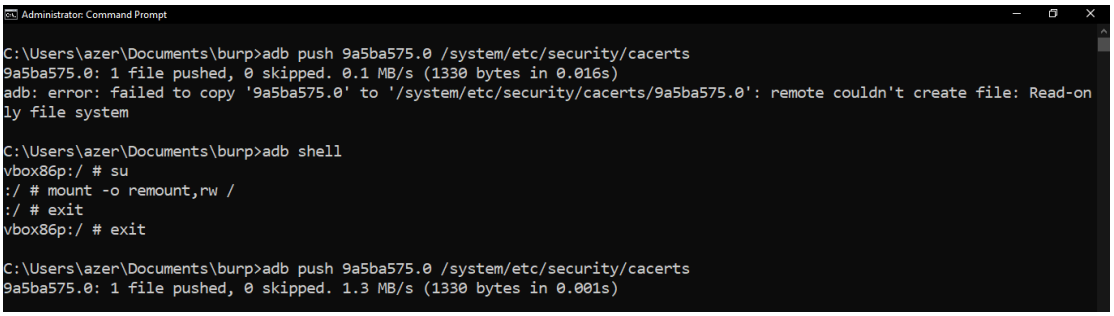
✓ Tout d'abord, nous devrons générer un certificat Burp Suite pour l'interception du trafic HTTPS.



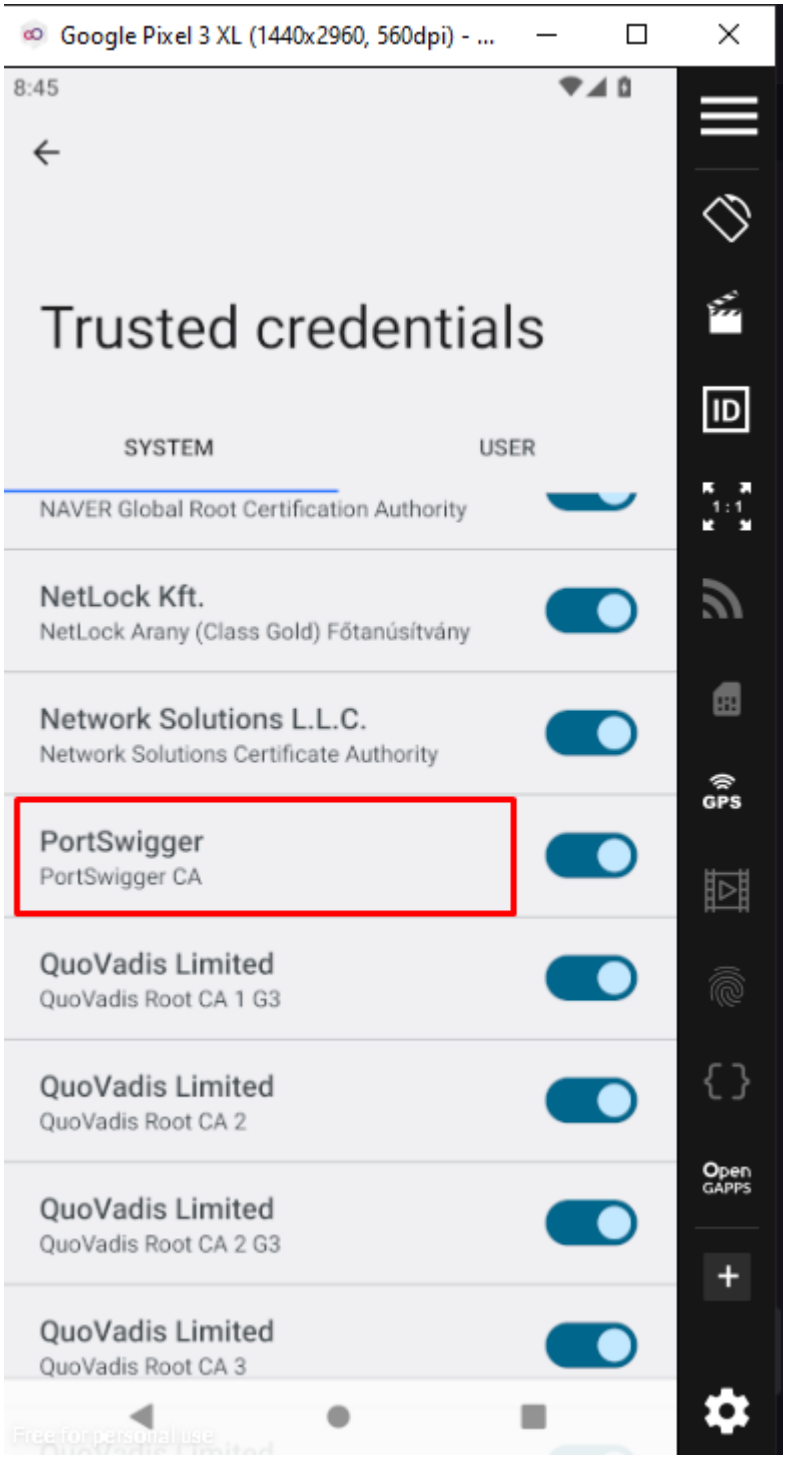
✓ en format der et puis on va utiliser l'outil OpenSsl pour la convertir en format PEM



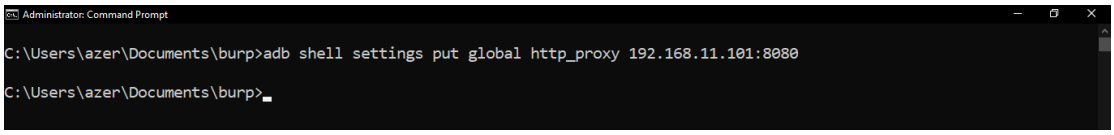
- ✓ On utilise la commande push d'adb afin de déplacer le fichier de la certification vers la sd card de l'émulateur.
- ✓ Mais nous n'avons pas la permission d'écrire dans le système de fichiers, d'abord nous allons le remonter en écriture et ensuite pousser le certificat à nouveau.



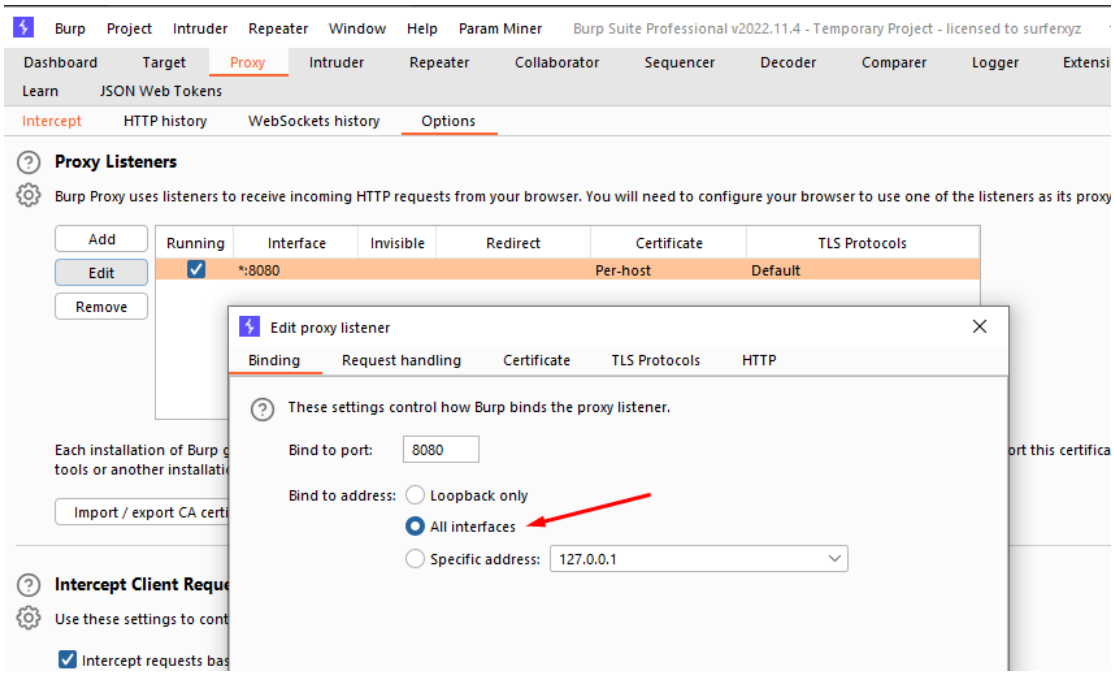
- ✓ Maintenant si on vérifie les certificats sur notre appareil émulé , on va retrouver celui de Burpsuite



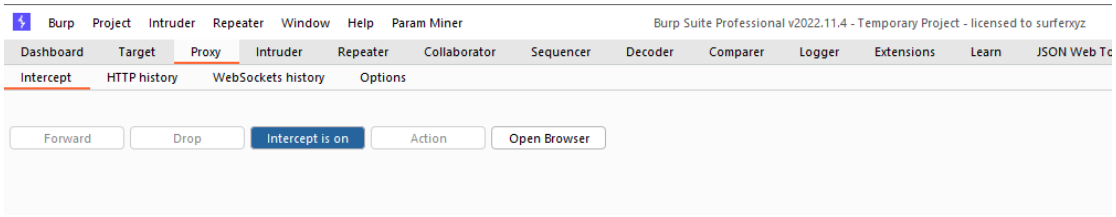
✓ Configurer le proxy via la ligne de commande.

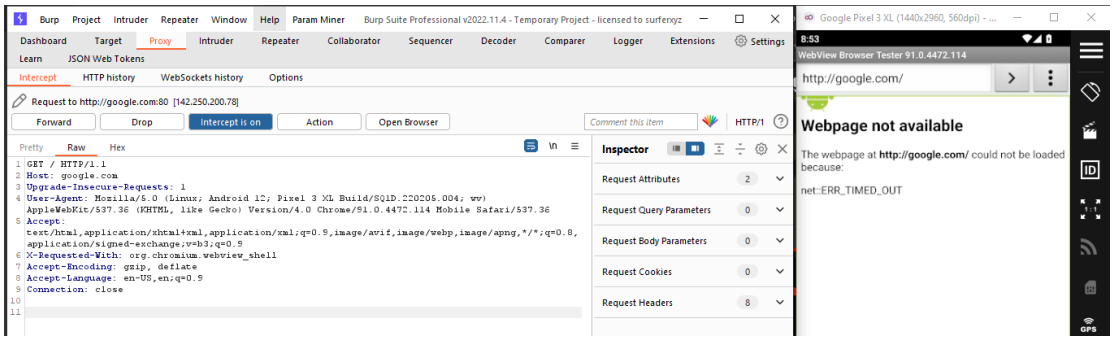


✓ Puisque Genymotion fonctionne comme une machine virtualisée, nous allons définir l'adresse **All interfaces** au lieu de **loopback** (127.0.0.1)

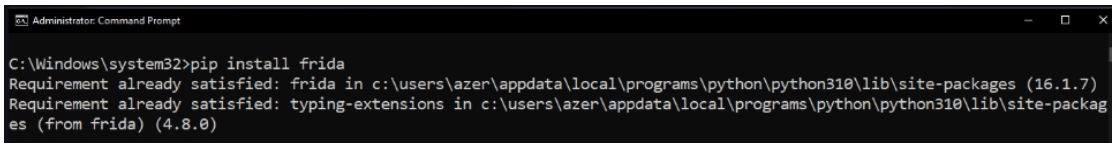


✓ Maintenant, si nous testons d'intercepter un quelconque trafic, nous remarquons que nous avons réussi.





✓ installing frida :

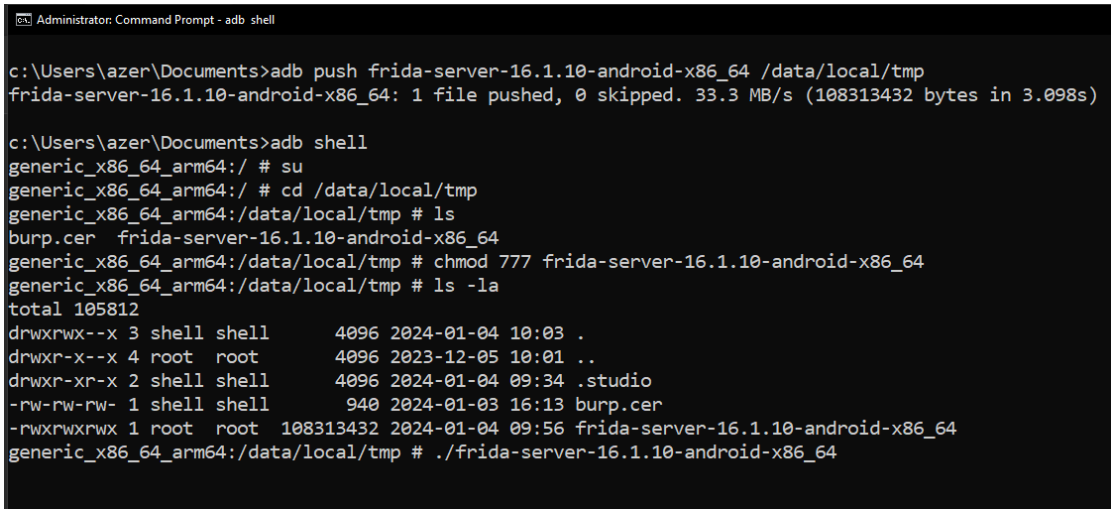


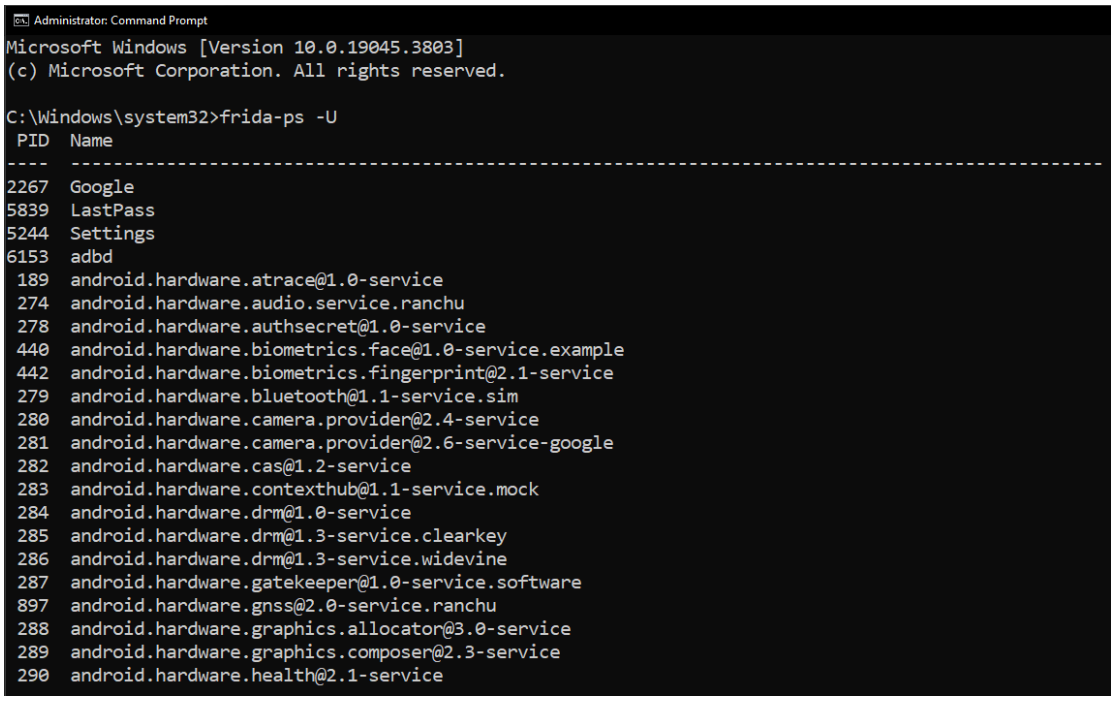

```
C:\Windows\system32>pip install frida-tools
Requirement already satisfied: frida-tools in c:\users\azer\appdata\local\programs\python\python310\lib\site-packages (12
.3.0)
Requirement already satisfied: colorama<1.0.0,>=0.2.7 in c:\users\azer\appdata\local\programs\python\python310\lib\site-p
ackages (from frida-tools) (0.4.6)
Requirement already satisfied: frida<17.0.0,>=16.0.9 in c:\users\azer\appdata\local\programs\python\python310\lib\site-pa
ckages (from frida-tools) (16.1.7)
Requirement already satisfied: prompt-toolkit<4.0.0,>=2.0.0 in c:\users\azer\appdata\local\programs\python\python310\lib\
site-packages (from frida-tools) (3.0.38)
Requirement already satisfied: pygments<3.0.0,>=2.0.2 in c:\users\azer\appdata\local\programs\python\python310\lib\site-p
ackages (from frida-tools) (2.14.0)
Requirement already satisfied: typing-extensions in c:\users\azer\appdata\local\programs\python\python310\lib\site-packag
es (from frida<17.0.0,>=16.0.9->frida-tools) (4.8.0)
Requirement already satisfied: wcwidth in c:\users\azer\appdata\local\programs\python\python310\lib\site-packages (from p
rompt-toolkit<4.0.0,>=2.0.0->frida-tools) (0.2.6)
```

✓ Obtenez l'architecture du CPU pour télécharger un serveur Frida compatible

```
C:\Windows\system32>adb shell
vbox86p:/ # su
:/ # uname -a
Linux localhost 5.10.101-genymotion+-ab74 #1 SMP PREEMPT Thu Dec 1 14:03:02 UTC 2022 x86_64
:/ #
```

✓ Execute le server frida



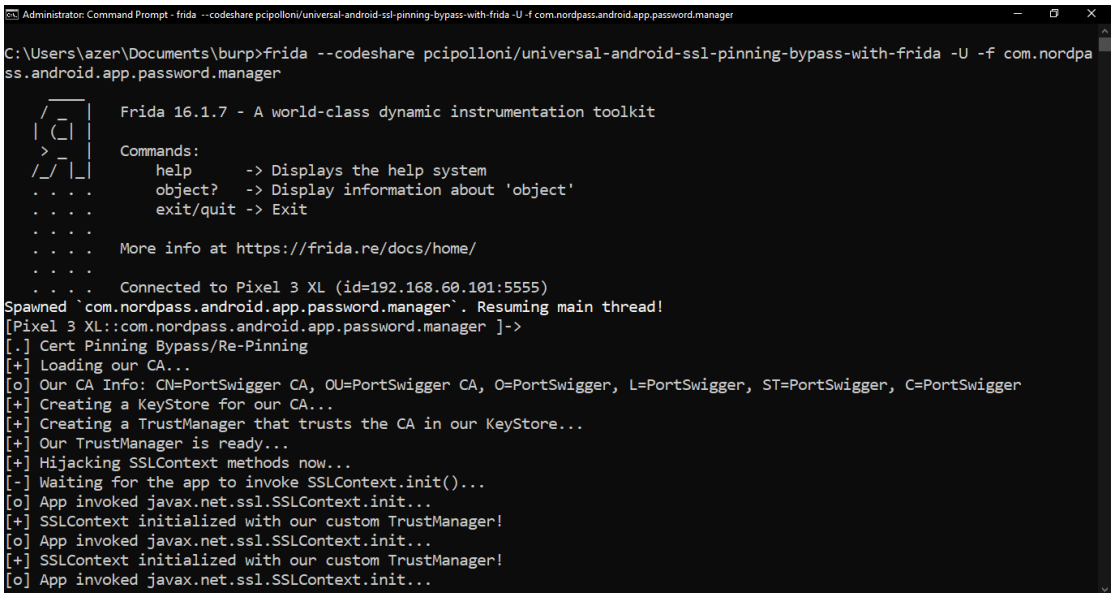


- SSL Pinning :

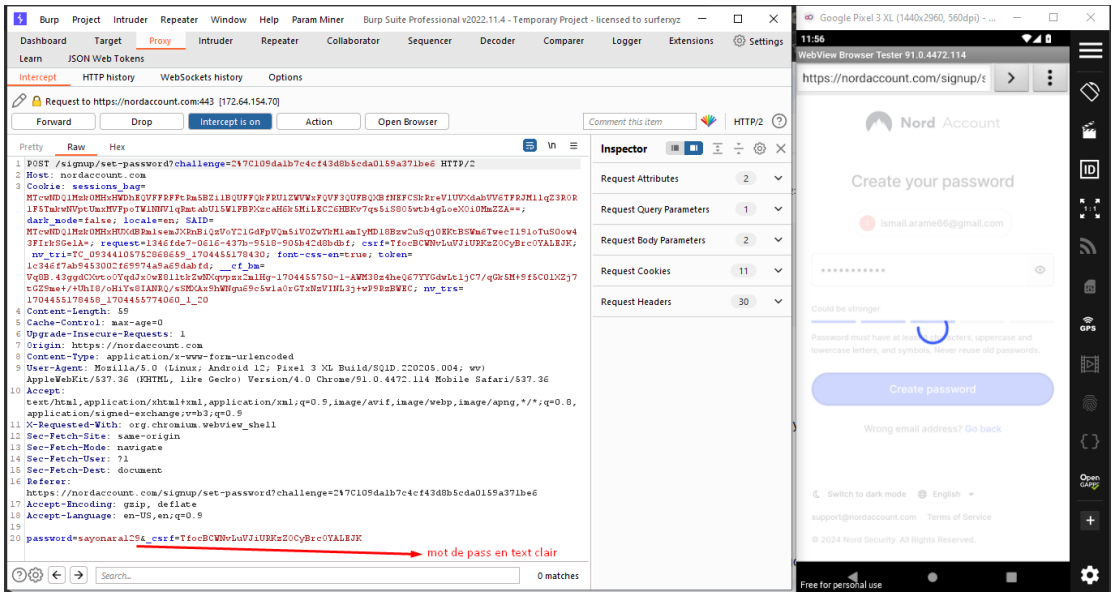
✓ On a besoin de certificat burpsuite pour le ssl pinning reussi

```
C:\Users\azer\Documents\burp>adb push burpcert.crt /data/local/tmp/cert-der.crt
burpcert.crt: 1 file pushed, 0 skipped. 0.7 MB/s (940 bytes in 0.001s)

C:\Users\azer\Documents\burp>adb shell
vbox86p:/ # su
:/ # cd /data/local/tmp
:/data/local/tmp # ls
cert-der.crt  frida-server-16.1.10-android-x86_64
:/data/local/tmp #
```

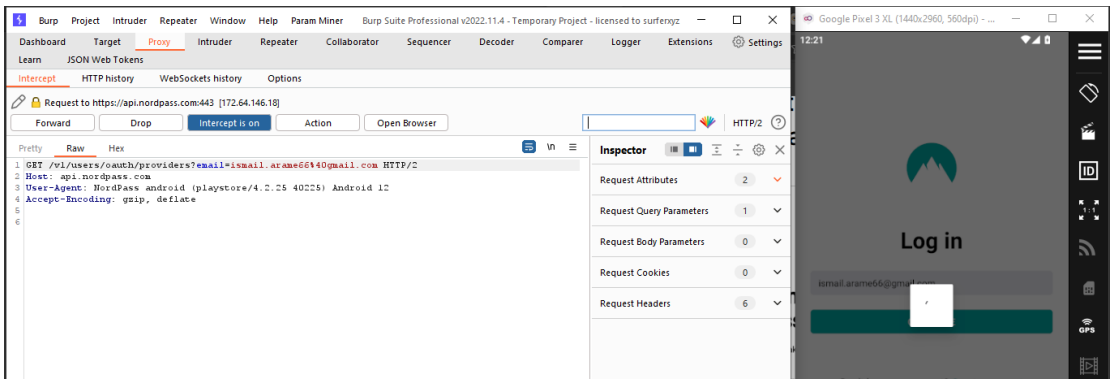
- ✓ Pour le SSL pinning bypass on trouve sur Frida Codeshare la commande nécessaire :
- ✓ https://codeshare.frida.re/@pcipolloni/universal-android-ssl-pinning-bypass-with-frida/



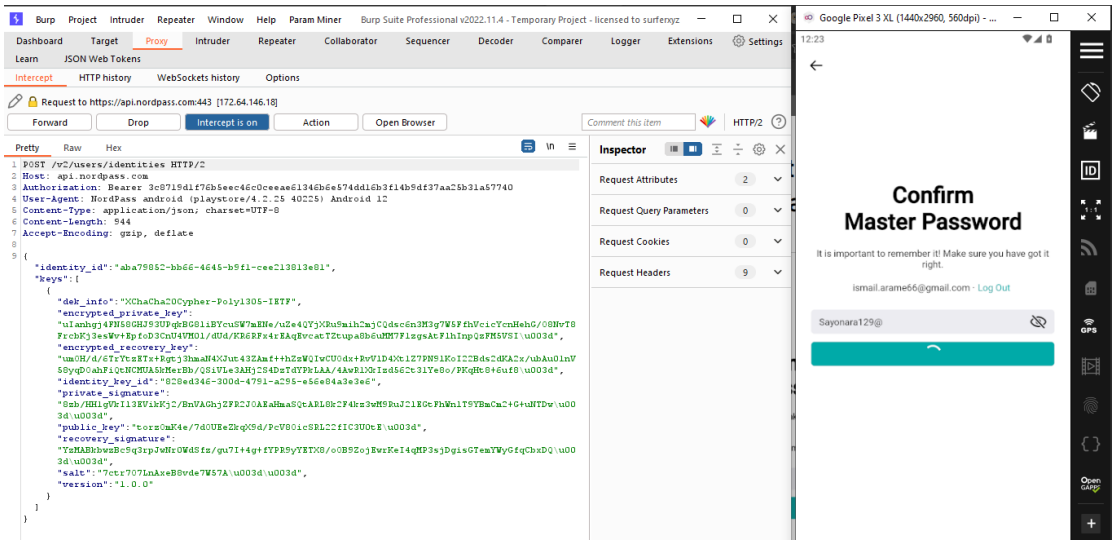
- ✓ Et on commance par la création de notre compte sur l'application : L'application me dérige vers leur page web pour créer un compte, j'ai entrer mes informations personnelles, apres l'envoie du requette au serveur j'ai remaqué que mon password est en text clair, d'où un attacker qui entrain d'espionner va avoir l'access à mon password



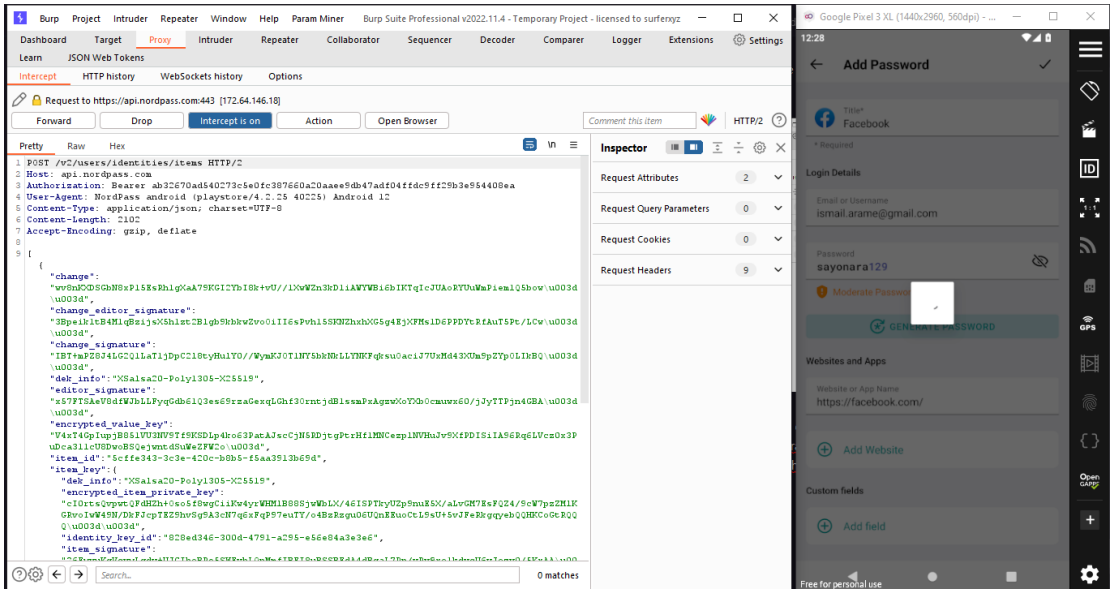
✓ Apres la création de mon compte sur website, j'ai passé au login sur l'application, et je vois encore que mon adresse émail est en texte clair lors de l'intecept du traffic :



✓ Ensuite NordPass me demande de créer un master password et le confirmer, et puis l'application m'a assigné un id et des clés publiques et privées chiffrées :

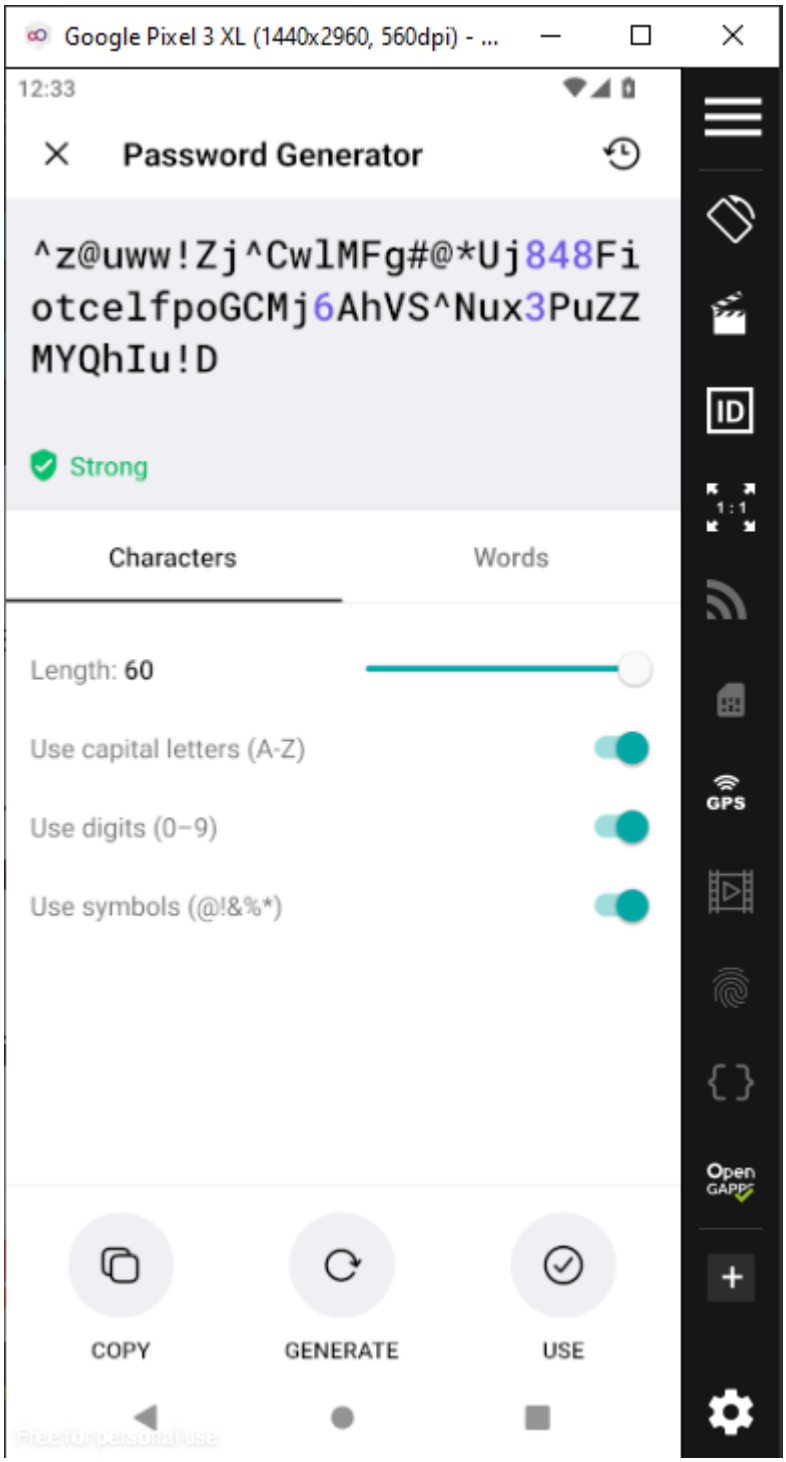


✓ D'outre, NordPass offre la possiblité de stocker les mots de passe des différentes applications. En titre d'exemple j'ai ajouté le mot de passe de mon compte facebook et aucune des informations que j'ai saisi est fachée, le traffic est chiffré avec des clés encore chiffrées :



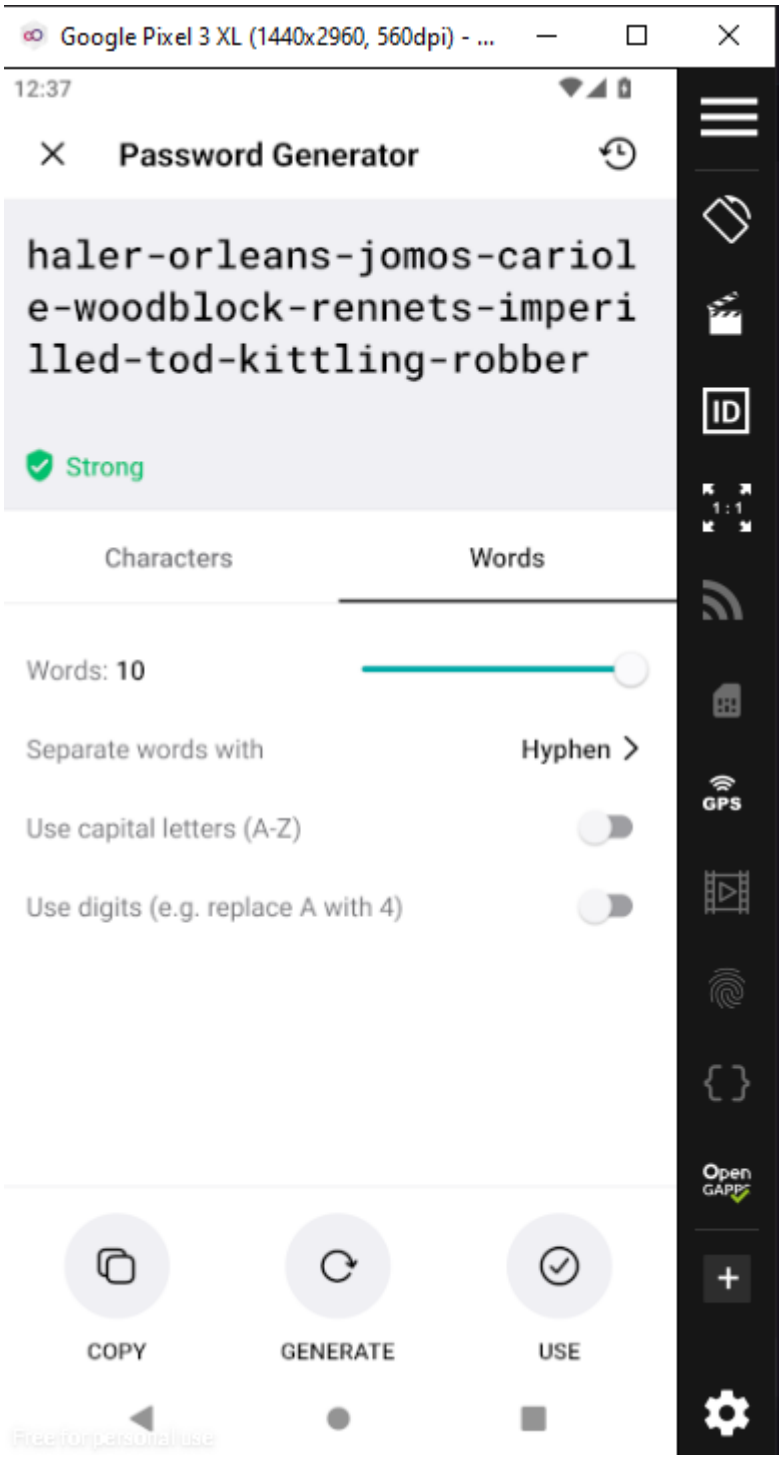
- Pour maintenir une sécurité de mot de passe uncrackable, NordPass fournit des générateurs de mots de passe robustes accessibles à tous, même sans abonnement.

- **NordPass** facilite la création de mots de passe hautement sécurisés d’une longueur maximale de 60 caractères, comprenant un mélange de lettres majuscules et minuscules, de chiffres et de symboles spéciaux.



✓ Il se distingue notamment par sa fonction de générateur de phrases de passe, permettant aux utilisateurs de créer des phrases de passe allant jusqu'à 10 mots, offrant une solution à

la fois sécurisée et plus facile à mémoriser par rapport à une série aléatoire de caractères.



➢ Keeper :

Static Analysis :

✓ Android.manifest.xml -

il y en a une seul activite qui peut etre execute par d'autre app.

AndroidManifest.xml

Find: exported="true"

```
249          <category android:name="android.intent.category.DEFAULT"/>
248      </intent-filter>
243  </activity>
252  <activity android:theme="@style/ImportPasswords" android:name="com.callpod.android_apps.keeper.onboarding.ImportPasswor
258      <intent-filter>
259          <category android:name="android.intent.category.DEFAULT"/>
258      </intent-filter>
252  </activity>
262  <activity android:theme="@style/Theme.Keeper.Light" android:label="@string/app_name" android:name="com.callpod.android_
268  <activity android:theme="@style/Theme.Keeper" android:name="com.callpod.android_apps.keeper.DetailLoadActivity" android
274      <intent-filter>
275          <action android:name="com.callpod.android_apps.keeper.DetailLoadActivity"/>
277          <category android:name="android.intent.category.DEFAULT"/>
274      </intent-filter>
268  </activity>
280  <activity android:theme="@android:style/Theme.NoDisplay" android:label="@string/app_name" android:name="com.callpod.and
284  <activity android:name="com.callpod.android_apps.keeper.ParseDeepLinkActivity" android:exported="true" android:launchMo
288      <intent-filter>
289          <action android:name="android.intent.action.VIEW"/>
291          <category android:name="android.intent.category.DEFAULT"/>
292          <category android:name="android.intent.category.BROWSABLE"/>
294          <data android:scheme="https" android:host="@string/deepLinkDomain" android:pathPrefix="/buy"/>
288      </intent-filter>
299      <intent-filter>
300          <action android:name="android.intent.action.VIEW"/>
302          <category android:name="android.intent.category.DEFAULT"/>
303          <category android:name="android.intent.category.BROWSABLE"/>
305          <data android:scheme="https" android:host="@string/deepLinkDomain" android:pathPrefix="/vault"/>
309          <data android:scheme="https" android:host="@string/euDeepLinkDomain" android:pathPrefix="/vault"/>
299      </intent-filter>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="572" android:versionName="15.0.1.1" and
    <uses-sdk android:minSdkVersion="23" android:targetSdkVersion="29"/>
    <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="tr
    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
    <uses-feature android:name="android.hardware.telephony" android:required="false"/>
    <uses-feature android:name="android.hardware.screen.portrait" android:required="false"/>
    <uses-feature android:name="android.hardware.touchscreen"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="com.android.vending.BILLING"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <uses-feature android:name="android.hardware.camera" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <application android:theme="@style/Theme.Keeper" android:label="Keeper" android:icon="@mipmap/ic_launcher" android:name="com.
        <meta-data android:name="com.google.android.gms.vision.DEPENDENCIES" android:value="barcode"/>
        <activity android:theme="@style/Theme.Keeper.Launcher" android:name="com.callpod.android_apps.keeper.ResultsActivity" and
```

✓ Resources.arc/res/values/strings:

On a trouve ce cle API :

```
34    <string name="get_verification_code_button">Send Code</string>
35    <string name="good_email_package">com.good.android.gfe</string>
36    <string name="google_api_key">AIzaSyCkTmHw3kL1rxdsQN71LinE24Vpmf2dWuI</string>
37    <string name="google_app_id">1:36874242427:android:701c34fe643c0444</string>
38    <string name="google_authenticator">Google and Microsoft Authenticator (TOTP)</string>
39    <string name="google_crash_reporting_api_key">AIzaSyCkTmHw3kL1rxdsQN71LinE24Vpmf2dWuI</string>
40    <string name="google_storage_bucket">graphite-wave-684.appspot.com</string>
41    <string name="got_it">Got It</string>
```

✓ URLS:

```
com.adjust.sdk.Constants                          public static final String BASE_URL = "https://app.adjust.com";
com.adjust.sdk.Constants                          public static final String GDPR_URL = "https://gdpr.adjust.com";
com.callpod.android_apps.keeper.common.BuildConf. public static final String KEEPERSECURITY_STATIC = "https://keepersecurity.com/android/";
com.callpod.android_apps.keeper.common.BuildConf. public static final String PROTOCOL = "https://";
com.callpod.android_apps.keeper.common.BuildConfi return new AutoValue_BuildConfigDecor.Builder().subDomain("").keeperHost(BuildConfig.KEEPER_HOST).apiRoot(BuildConfig.KEEPE
com.callpod.android_apps.keeper.common.util.Strin if (replaceAll.startsWith("https://") || replaceAll.startsWith("http://")) {
com.callpod.android_apps.keeper.common.util.Strin return "https://" + replaceAll;
com.google.android.gms.auth.api.credentials.Iden. public static final String FACEBOOK = "https://www.facebook.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String GOOGLE = "https://accounts.google.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String LINKEDIN = "https://www.linkedin.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String MICROSOFT = "https://login.live.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String PAYPAL = "https://www.paypal.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String TWITTER = "https://twitter.com";
com.google.android.gms.auth.api.credentials.Iden. public static final String YAHOO = "https://login.yahoo.com";
com.google.android.gms.common.FirstPartyScopes    public static final String GAMES_1P = "https://www.googleapis.com/auth/games.firstparty";
com.google.android.gms.common.Scopes              public static final String APP_STATE = "https://www.googleapis.com/auth/appstate";
com.google.android.gms.common.Scopes              public static final String CLOUD_SAVE = "https://www.googleapis.com/auth/datastoremobile";
com.google.android.gms.common.Scopes              public static final String DRIVE_APPFOLDER = "https://www.googleapis.com/auth/drive.appdata";
com.google.android.gms.common.Scopes              public static final String DRIVE_APPS = "https://www.googleapis.com/auth/drive.apps";
com.google.android.gms.common.Scopes              public static final String DRIVE_FILE = "https://www.googleapis.com/auth/drive.file";
com.google.android.gms.common.Scopes              public static final String DRIVE_FULL = "https://www.googleapis.com/auth/drive";
com.google.android.gms.common.Scopes              public static final String FITNESS_ACTIVITY_READ = "https://www.googleapis.com/auth/fitness.activity.read";
com.google.android.gms.common.Scopes              public static final String FITNESS_ACTIVITY_READ_WRITE = "https://www.googleapis.com/auth/fitness.activity.write";
com.google.android.gms.common.Scopes              public static final String FITNESS_BLOOD_GLUCOSE_READ = "https://www.googleapis.com/auth/fitness.blood_glucose.read";
com.google.android.gms.common.Scopes              public static final String FITNESS_BLOOD_GLUCOSE_READ_WRITE = "https://www.googleapis.com/auth/fitness.blood_glucose.write";
com.google.android.gms.common.Scopes              public static final String FITNESS_BLOOD_PRESSURE_READ = "https://www.googleapis.com/auth/fitness.blood_pressure.read";
```

✓ Analyse du code :

dans com.callpod.android_apps.keeper.common.password

```java
package com.callpod.android_apps.keeper.common.password;

import java.security.SecureRandom;
import kotlin.Metadata;
import kotlin.collections.CollectionsKt;
import kotlin.jvm.internal.DefaultConstructorMarker;
import kotlin.jvm.internal.Intrinsics;
import kotlin.sequences.SequencesKt;
import kotlin.text.StringsKt;

/* compiled from: EvenlyDistributedPasswordGenerator.kt */
@Metadata(bv = {1, 0, 3}, d1 = {"\u0000*\n\u0002\u0018\u0002\n\u0002\u0018\u0002\n\u0002\b\u0002\n\u0002
/* loaded from: classes2.dex */
public final class EvenlyDistributedPasswordGenerator implements PasswordStrategy {
    private static final Companion Companion = new Companion(null);
    private static final String TAG;
    private static final String digits = "0123456789";
    private static final String lowerCase = "abcdefghijklmnopqrstuvwxyz";
    private static final int maxPasswordLen = 200;
    private static final int minPasswordLen = 4;
    private static final String symbols = "!@#$%";
    private static final String upperCase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    private PasswordConfig config;
    private final SecureRandom secureRandom = new SecureRandom();

    /* compiled from: EvenlyDistributedPasswordGenerator.kt */
    @Metadata(bv = {1, 0, 3}, d1 = {"\u0000\u001c\n\u0002\u0018\u0002\n\u0002\u0010\u0000\n\u0002\b\u000
    /* loaded from: classes2.dex */
    private static final class Companion {
        private Companion() {
```

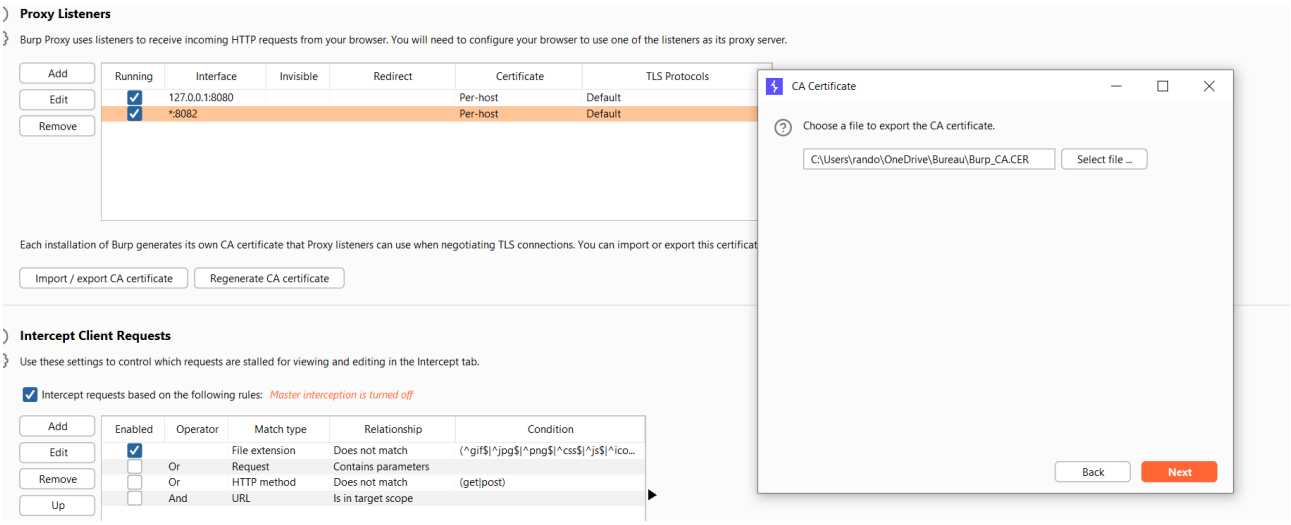
- on trouve la fonction ***generateRandomizedPassword().***

```java
    private final String generateRandomizedPassword() {
        PasswordConfig passwordConfig = this.config;
        if (passwordConfig == null) {
            Intrinsics.throwUninitializedPropertyAccessException("config");
        }
        int max = Math.max(Math.min(passwordConfig.getLength(), 200), 4);
        PasswordConfig passwordConfig2 = this.config;
        if (passwordConfig2 == null) {
            Intrinsics.throwUninitializedPropertyAccessException("config");
        }
        int useDigitsMin = passwordConfig2.getUseDigitsMin();
        PasswordConfig passwordConfig3 = this.config;
        if (passwordConfig3 == null) {
            Intrinsics.throwUninitializedPropertyAccessException("config");
        }
        int useUpperCaseMin = useDigitsMin + passwordConfig3.getUseUpperCaseMin();
        PasswordConfig passwordConfig4 = this.config;
        if (passwordConfig4 == null) {
            Intrinsics.throwUninitializedPropertyAccessException("config");
        }
        int useSymbolsMin = useUpperCaseMin + passwordConfig4.getUseSymbolsMin();
        PasswordConfig passwordConfig5 = this.config;
        if (passwordConfig5 == null) {
```

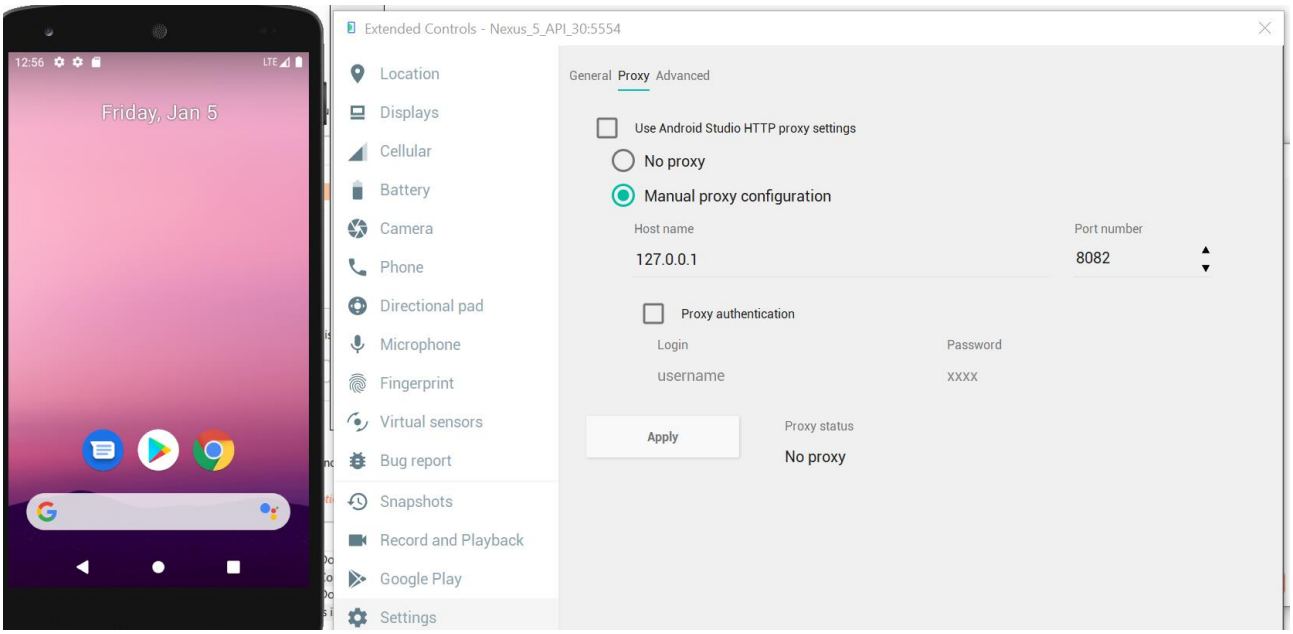
- path: **androidx.security.crypto**

```java
/* loaded from: classes.dex */
public final class MasterKeys {
    private static final String ANDROID_KEYSTORE = "AndroidKeyStore";
    static final String KEYSTORE_PATH_URI = "android-keystore://";
    private static final int KEY_SIZE = 256;
    static final String MASTER_KEY_ALIAS = "_androidx_security_master_key_";
    public static final KeyGenParameterSpec AES256_GCM_SPEC = createAES256GCMKeyGenParameterSpec(MASTER_KEY_ALIAS);

    private MasterKeys() {
    }

    private static KeyGenParameterSpec createAES256GCMKeyGenParameterSpec(String str) {
        return new KeyGenParameterSpec.Builder(str, 3).setBlockModes(CodePackage.GCM).setEncryptionPaddings("NoPadding").setKeySize(256).build();
    }

    public static String getOrCreate(KeyGenParameterSpec keyGenParameterSpec) throws GeneralSecurityException, IOException {
        validate(keyGenParameterSpec);
        if (!keyExists(keyGenParameterSpec.getKeystoreAlias())) {
            generateKey(keyGenParameterSpec);
        }
        return keyGenParameterSpec.getKeystoreAlias();
    }

    static void validate(KeyGenParameterSpec keyGenParameterSpec) {
        if (keyGenParameterSpec.getKeySize() != 256) {
            throw new IllegalArgumentException("invalid key size, want 256 bits got " + keyGenParameterSpec.getKeySize() + " bits");
        } else if (!Arrays.equals(keyGenParameterSpec.getBlockModes(), new String[]{CodePackage.GCM})) {
            throw new IllegalArgumentException("invalid block mode, want GCM got " + Arrays.toString(keyGenParameterSpec.getBlockModes()));
        } else if (keyGenParameterSpec.getPurposes() != 3) {
            throw new IllegalArgumentException("invalid purposes mode, want PURPOSE_ENCRYPT | PURPOSE_DECRYPT got " + keyGenParameterSpec.getPurposes());
        } else if (!Arrays.equals(keyGenParameterSpec.getEncryptionPaddings(), new String[]{"NoPadding"})) {
            throw new IllegalArgumentException("invalid padding mode, want NoPadding got " + Arrays.toString(keyGenParameterSpec.getEncryptionPaddings()));
        } else if (keyGenParameterSpec.isUserAuthenticationRequired() && keyGenParameterSpec.getUserAuthenticationValidityDurationSeconds() < 1) {
            throw new IllegalArgumentException("per-operation authentication is not supported (UserAuthenticationValidityDurationSeconds must be >0)");
        }
    }

    private static void generateKey(KeyGenParameterSpec keyGenParameterSpec) throws GeneralSecurityException {
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES", ANDROID_KEYSTORE);
        keyGenerator.init(keyGenParameterSpec);
        keyGenerator.generateKey();
    }
```

Dynamic analysis :



✓ **Créer un listener**



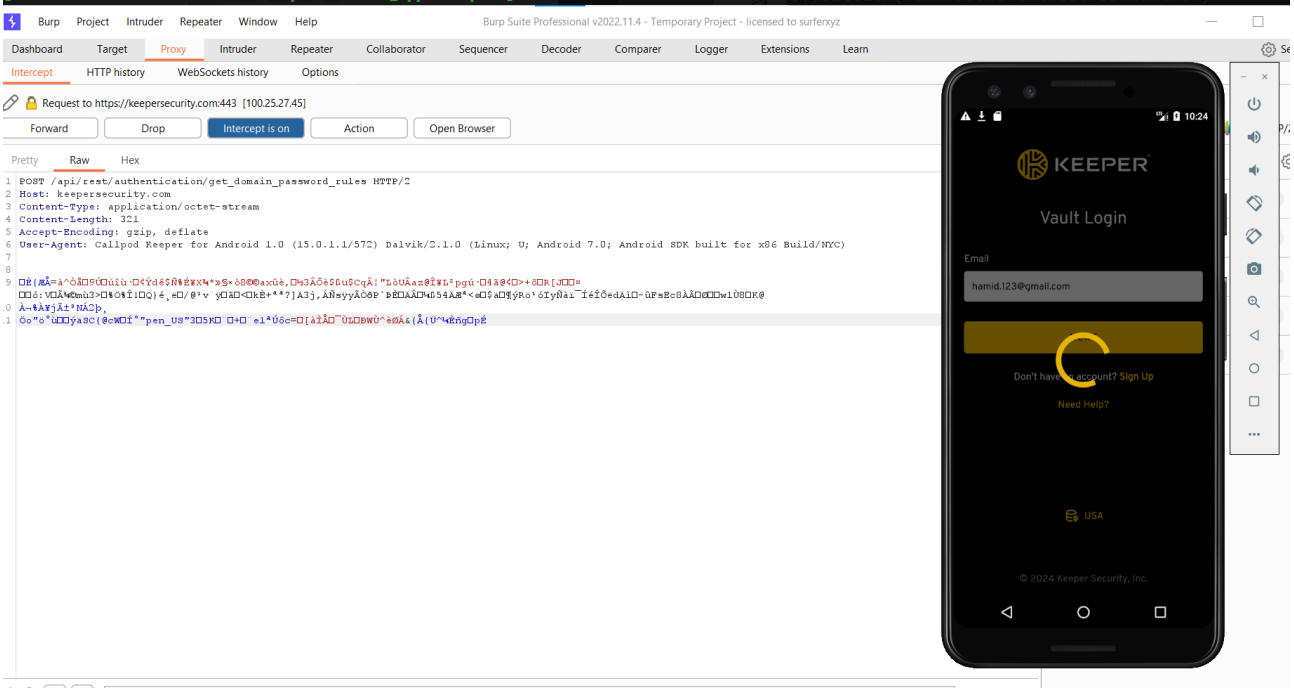
✓ **Injection et execution de frida-server:**

```
C:\Users\rando\OneDrive\Bureau>py -m pip install frida
Collecting frida
  Downloading frida-16.1.10-cp37-abi3-win_amd64.whl (32.6 MB)
     ---------------------------------------- 32.6/32.6 MB 438.7 kB/s eta 0:00:00
Installing collected packages: frida
Successfully installed frida-16.1.10

[notice] A new release of pip available: 22.3.1 -> 23.3.2
[notice] To update, run: C:\Users\rando\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip

C:\Users\rando\OneDrive\Bureau>py -m pip install objection
Collecting objection
  Downloading objection-1.11.0.tar.gz (327 kB)
     ---------------------------------------- 327.2/327.2 kB 53.1 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: frida>=14.0.0 in c:\users\rando\appdata\local\programs\python\python311\lib\site-packages (from objection) (
Collecting frida-tools>=6.0.0
  Downloading frida-tools-12.3.0.tar.gz (200 kB)
     ---------------------------------------- 200.5/200.5 kB 81.2 kB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting prompt_toolkit<4.0.0,>=3.0.3
  Downloading prompt_toolkit-3.0.43-py3-none-any.whl (386 kB)
     ---------------------------------------- 386.1/386.1 kB 250.6 kB/s eta 0:00:00
```

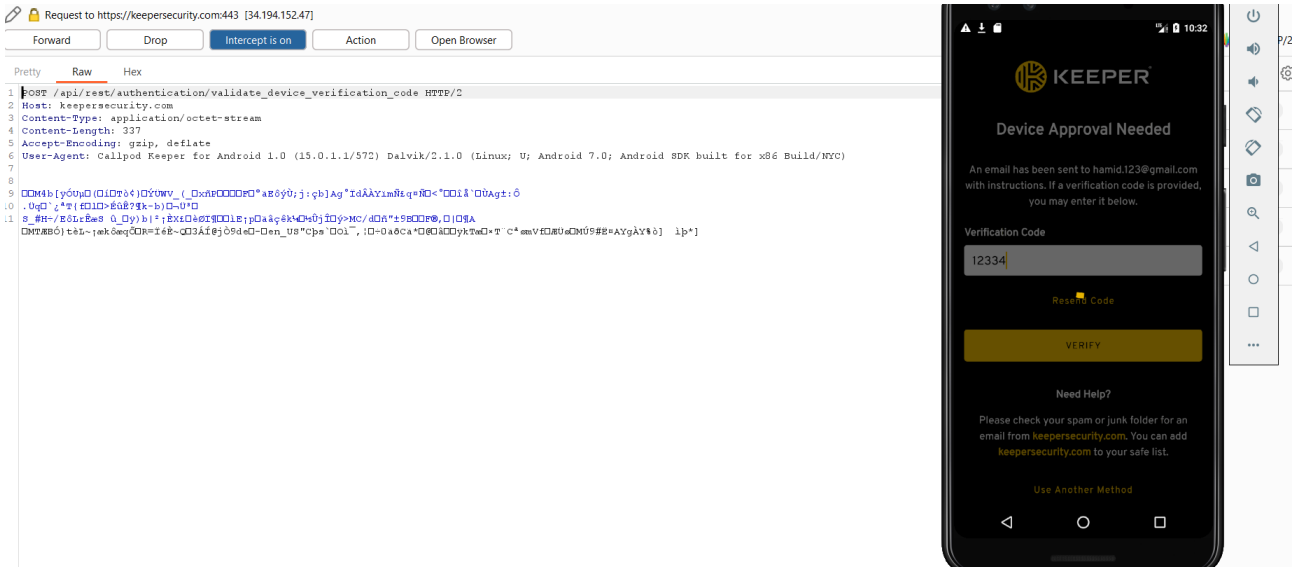
```
C:\Users\rando\Downloads>adb push "C:\Users\rando\Downloads\frida-server-16.1.10-android-x86\frida-server-16.1.10-android-x86" data/local/tmp
C:\Users\rando\Downloads\frida-server-16.1.10-android-x86\frida-server-16.1.10-android-x86: 1 file pushed, 0 skipped. 16.8 MB/s (51684860 bytes in 2.926s)

C:\Users\rando\Downloads>adb shell "chmod +x data/local/tmp/frida-server-16.1.10-android-x86"

C:\Users\rando\Downloads>adb shell "data/local/tmp/frida-server-16.1.10-android-x86 &"
```

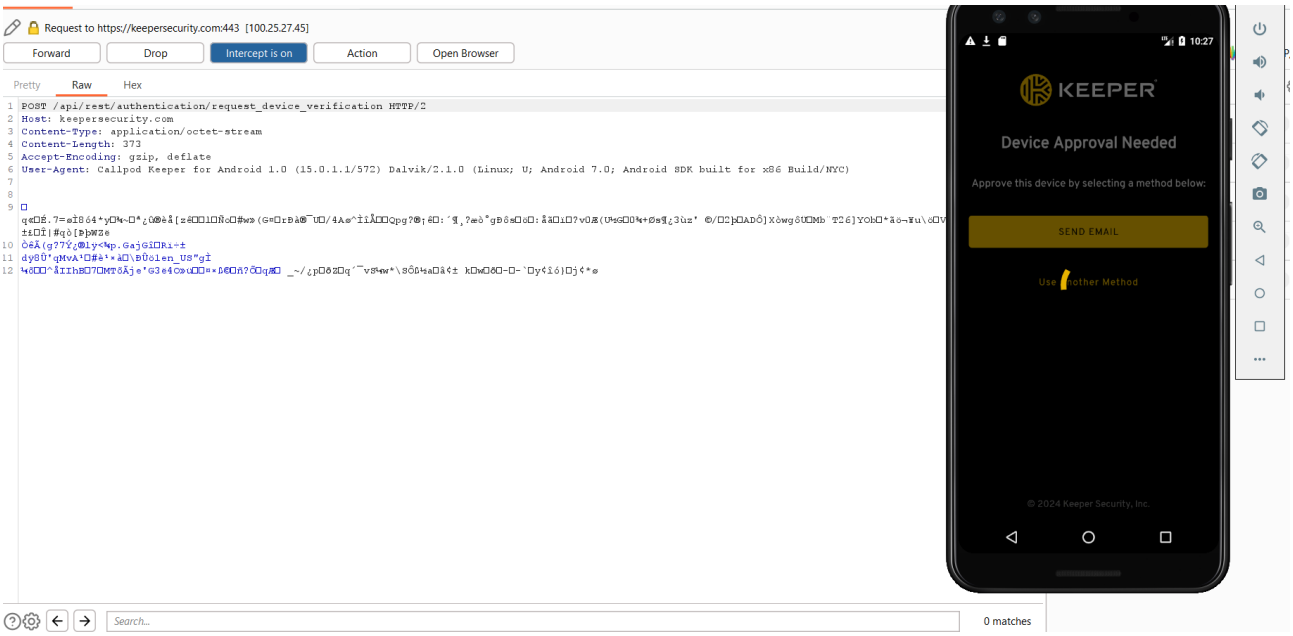
✓ Bypass SSL pinning

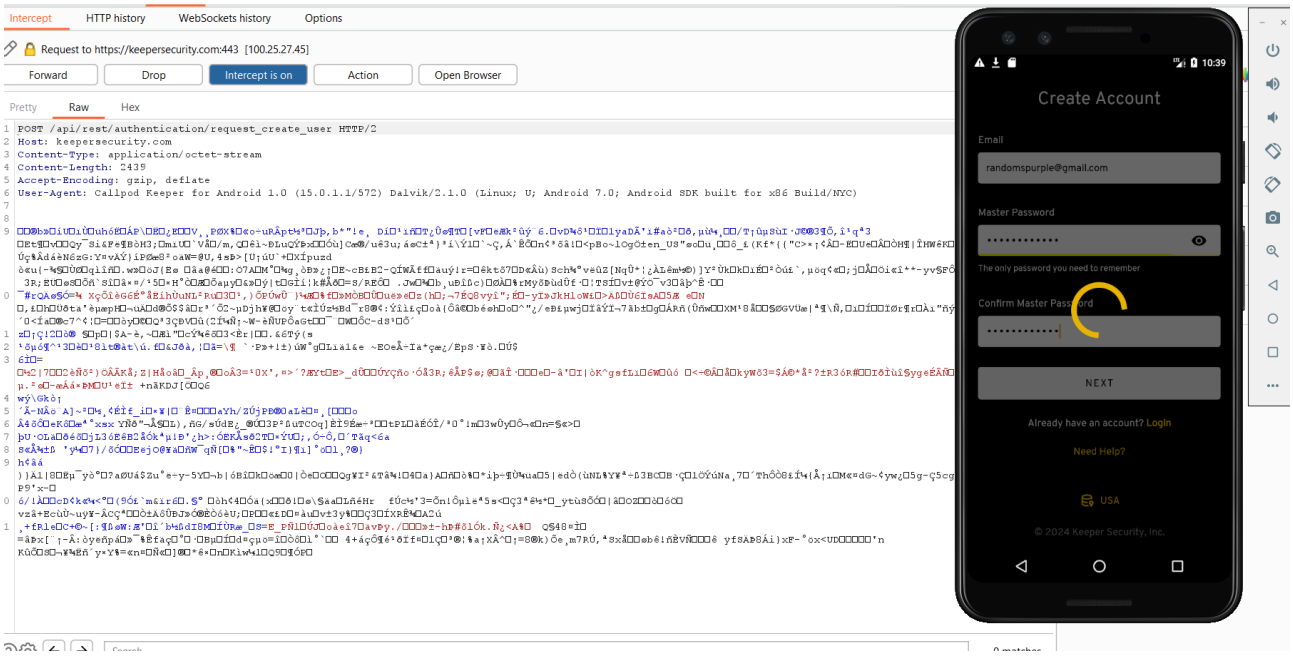
```
C:\Users\rando\Downloads>frida --codeshare pcipolloni/universal-android-ssl-pinning-bypass-with-frida -U -f com.callpod.android_apps.keeper
     ____
    / _  |   Frida 16.1.10 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .   Connected to Android Emulator 5554 (id=emulator-5554)

Spawned `com.callpod.android_apps.keeper`. Resuming main thread!
[Android Emulator 5554::com.callpod.android_apps.keeper ]->
[.] Cert Pinning Bypass/Re-Pinning
[+] Loading our CA...
[o] Our CA Info: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
[+] Creating a KeyStore for our CA...
[+] Creating a TrustManager that trusts the CA in our KeyStore...
[+] Our TrustManager is ready...
[+] Hijacking SSLContext methods now...
[-] Waiting for the app to invoke SSLContext.init()...
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
[Android Emulator 5554::com.callpod.android_apps.keeper ]->
```



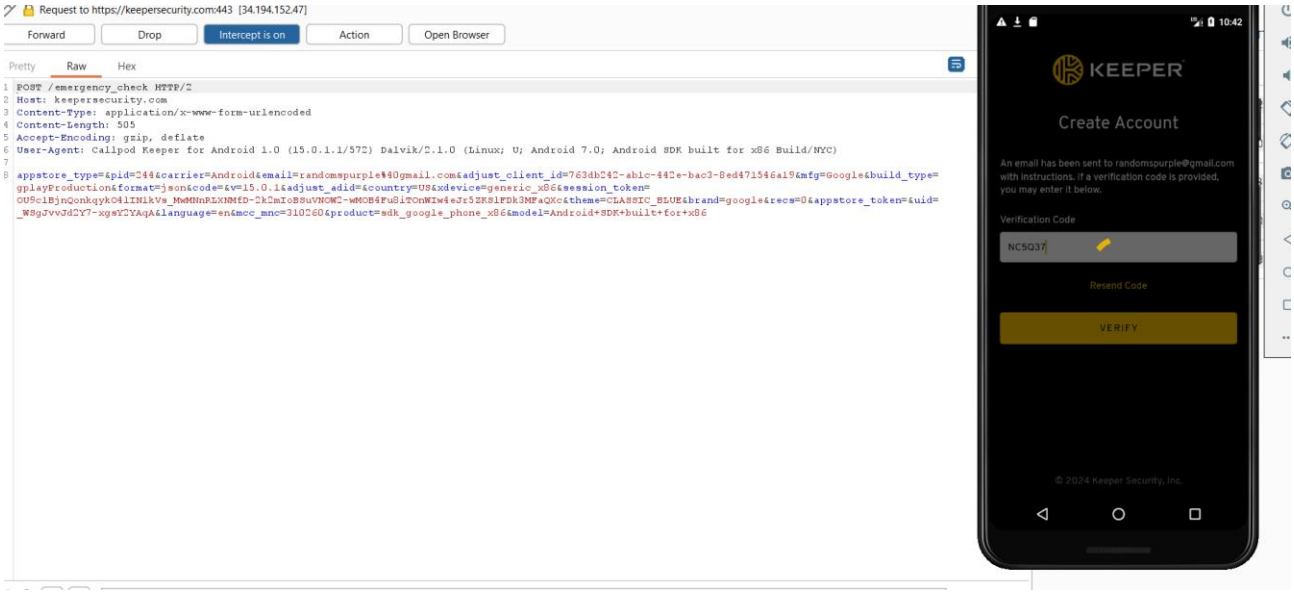
✓ **Voila le traffic apres le SSL pinning bypass**

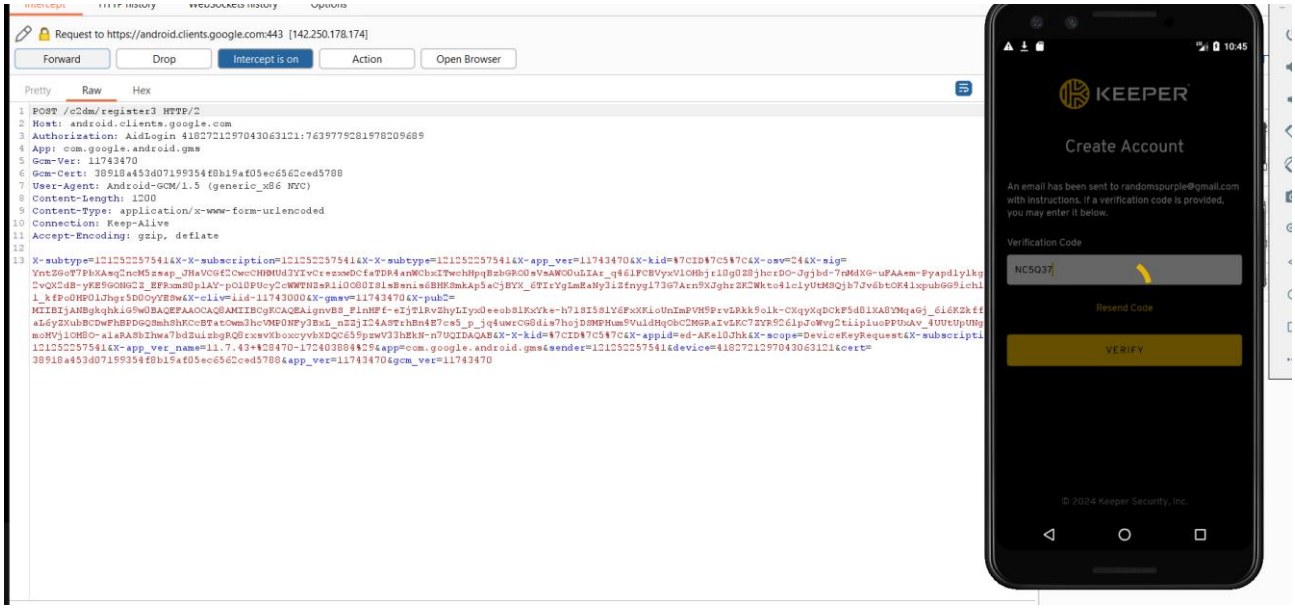




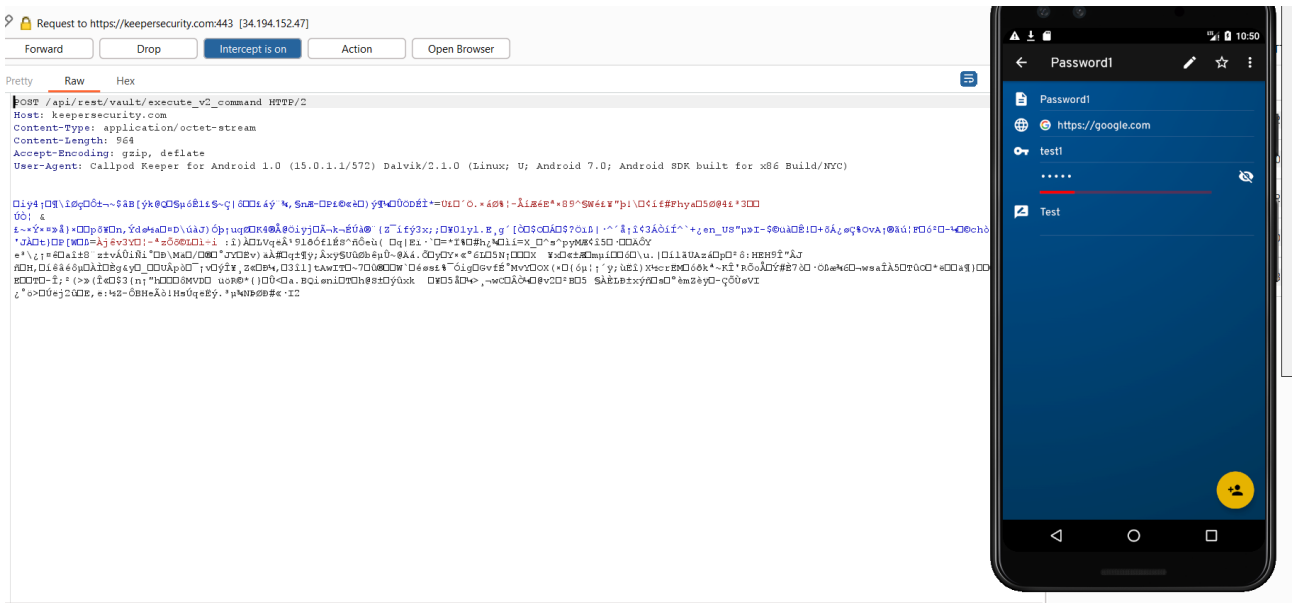


- *Deux requets peut extre exploite par un attacker **(MITM attack)** car les donnés ne sont pas chiffré:*





j'ai essayé d'ajouter un website sur l'application



➢ mSecure :

Static analysis :

✓ Androidmanifest.xml:

il y en a un seul **content provider** qui peut etre utiliser par les autres app

```
<activity android:name="com.mseven.msecure.ItemInfo" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.NewField" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.PasswordValidation" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale" android:windowSoftInputMode=
<activity android:name="com.mseven.msecure.Login" android:configChanges="keyboardHidden|keyboard|locale" android:windowSoftInputMode="stateAlwaysVisible"/>
<activity android:name="com.mseven.msecure.About" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.PasswordGenerator" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<receiver android:name="com.mseven.msecure.BootService">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</receiver>
<activity android:name="com.mseven.msecure.ReorderFields" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name="com.mseven.msecure.EditConnection" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.SelectConnection" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.GmailSetting" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.SendBackupEmail" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.SyncPassword" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.GetSupport" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.CopyToClipboard" android:launchMode="singleInstance" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|loc
<activity android:name="com.mseven.msecure.mBrowser"/>
<activity android:name="com.mseven.msecure.mBrowserTabs" android:launchMode="singleTask" android:configChanges="screenSize|orientation|keyboardHidden"/>
<activity android:name="com.mseven.msecure.ForcedOverflowItem" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.dropbox.client2.android.AuthActivity" android:launchMode="singleTask" android:configChanges="screenSize|orientation|keyboard">
    <intent-filter>
        <data android:scheme="db-mhh1ubyxbnwfhv0"/>
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.BROWSABLE"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
<activity android:name="com.mseven.msecure.WelcomeScreen" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.cropimage.CropImage"/>
<activity android:name="com.mseven.msecure.RecoverDeletedItems" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<activity android:name="com.mseven.msecure.DisplayImage" android:configChanges="screenSize|orientation|keyboardHidden|keyboard|locale"/>
<provider android:name="InternalStorageContentProvider" android:exported="true" android:authorities="com.mseven.msecure"/>
<activity android:name="a.y.z.A" android:configChanges="screenSize|screenLayout|orientation|keyboardHidden"/>
<meta-data android:name="default-activity" android:value="com.mseven.msecure.Licensing"/>
<activity android:name="a.y.z.HZ" android:configChanges="screenSize|screenLayout|orientation|keyboardHidden|keyboard|locale" android:windowSoftInputMode="stateAlwaysHidd
    <intent-filter>
        <category android:name="android.intent.category.LAUNCHER"/>
        <action android:name="android.intent.action.MAIN"/>
    </intent-filter>
</activity>
</application>
```

✓ META-INF/MANIFEST.MF :

```
1  Manifest-Version: 1.0
2  Created-By: 1.8.0_45 (Oracle Corporation)
3
4  Name: res/drawable-xhdpi-v4/abs__progress_secondary_holo_dark.9.png
5  SHA1-Digest: D0QKteBrrHhkWTE9BB+d8/oINbI=
6
7  Name: res/drawable-xxhdpi-v4/ic_action_add.png
8  SHA1-Digest: KJMeq5pZx/JncZi0JPmXcdcrKD8=
9
10 Name: res/drawable/popup_side_img.png
11 SHA1-Digest: CsI8fLPh3MN5pQjniZ6+r3ZyLYU=
12
13 Name: res/drawable-mdpi-v4/abs__ab_solid_light_holo.9.png
14 SHA1-Digest: qPBmGe92FV9luR4XdeDll8Xokio=
15
16 Name: res/drawable/add_tab.png
17 SHA1-Digest: JfmbDUDv3PL5Id83v14ftxWf6fA=
18
19 Name: res/drawable-nodpi-v4/music2.png
20 SHA1-Digest: 2P9hoxDUfi6X8dS8ix+9e+q+Q1g=
21
22 Name: res/drawable-nodpi-v4/note.png
23 SHA1-Digest: TrQ4nxfhgqOs32WRdLUPBe2cNPc=
24
25 Name: res/layout/cropimage.xml
26 SHA1-Digest: bFbQip7Ld50ZEKX/Q47U33jHR2A=
27
28 Name: res/drawable-xxhdpi-v4/ic_menu_settings_white.png
29 SHA1-Digest: aokNntwCTsOBVa3RmO/4W3kTBt4=
30
31 Name: res/drawable/menu_trash.png
32 SHA1-Digest: W62qzQu/qwnkeIKlDi3KJu0n/18=
33
34 Name: res/layout/editgroup.xml
35 SHA1-Digest: urHj4VKto5WKubhgjRZX4u642HA=
36
37 Name: res/drawable-nodpi-v4/web_violet.png
38 SHA1-Digest: BH/H9Yk9OHnUGQJmlGVJBcy7Cis=
39
40 Name: res/drawable/customdrawablecheckbox.xml
41 SHA1-Digest: P2UZIYRa+L0Oh5e/P1hr7pGIobc=
42
43 Name: res/drawable-nodpi-v4/burger_template.png
44 SHA1-Digest: ywWBkq5Fk7JnQhTc43H/7zBPVyk=
```

✓ Res/values/strings.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="abs__action_bar_home_description">Navigate home</string>
    <string name="abs__action_bar_up_description">Navigate up</string>
    <string name="abs__action_menu_overflow_description">More options</string>
    <string name="abs__action_mode_done">Done</string>
    <string name="abs__activity_chooser_view_see_all">See all...</string>
    <string name="abs__activity_chooser_view_dialog_title_default">Select activity</string>
    <string name="abs__share_action_provider_share_with">Share with...</string>
    <string name="abs__activitychooserview_choose_application">Choose an application</string>
    <string name="abs__shareactionprovider_share_with">Share with</string>
    <string name="abs__shareactionprovider_share_with_application">Share with %s</string>
    <string name="dlg_msg_payment_success">The payment was successful</string>
    <string name="IDS_SAPPS_POP_AN_ERROR_OCCURRED_WHILE_RESETTING_SAMSUNG_IN_APP_PURCHASE">An error occurred while resetting Samsung In-App Purchase</string>
    <string name="IDS_SAPPS_POP_AN_INVALID_VALUE_HAS_BEEN_PROVIDED_FOR_SAMSUNG_IN_APP_PURCHASE">An invalid value has been provided for Samsung In-App Purchase</string>
    <string name="IDS_SAPPS_POP_AN_UNEXPECTED_ERROR_HAS_OCCURRED_SAMSUNG_ACCOUNT_AUTHENTICATION_HAS_BEEN_CANCELLED">An unexpected error has occurred. Samsung account authentic
    <string name="IDS_SAPPS_POP_YOUR_PURCHASE_VIA_SAMSUNG_IN_APP_PURCHASE_IS_INVALID_A_FAKE_APPLICATION_HAS_BEEN_DETECTED_CHECK_THE_APP_MSG">Your purchase via Samsung In-App P
    <string name="IDS_SAPPS_POP_UNKNOWN_ERROR_OCCURRED">Unknown error occurred</string>
    <string name="IDS_SAPPS_POP_SAMSUNG_IN_APP_PURCHASE">Samsung In-App Purchase</string>
    <string name="IDS_SAPPS_POP_SAMSUNG_IN_APP_PURCHASE_NEEDS_TO_BE_UPDATED">Samsung In-App Purchase needs to be updated</string>
    <string name="IDS_SAPPS_POP_TO_PURCHASE_ITEMS_YOU_NEED_TO_INSTALL_SAMSUNG_IN_APP_PURCHASE_INSTALL_Q">To purchase items, you need to install Samsung In-App Purchase. Instal
    <string name="IDS_SAPPS_POP_A_NEW_VERSION_OF_SAMSUNG_IN_APP_PURCHASE_IS_AVAILABLE_UPDATE_Q">A new version of Samsung In-App Purchase is available. Update?</string>
    <string name="IDS_SAPPS_BODY_WAITING_ING">Waiting...</string>
    <string name="IDS_SAPPS_POP_PRODUCT_DOES_NOT_EXIST_IN_THIS_STORE">Product does not exist in this store</string>
    <string name="IDS_SAPPS_POP_ALREADY_PURCHASED">Already purchased</string>
    <string name="IDS_SAPPS_POP_PAYMENT_CANCELLED">Payment cancelled</string>
    <string name="IDS_SAPPS_POP_NETWORK_UNAVAILABLE">Network not available</string>
    <string name="IDS_SAPPS_HEADER_ITEM_PURCHASED">Item purchased</string>
    <string name="delQunT">Are you sure you want to delete this type?</string>
    <string name="errorStr">Error</string>
    <string name="cancelStr">Cancel</string>
    <string name="GmailCancelBtnStr">Cancel</string>
    <string name="passgenCancelText">Cancel</string>
    <string name="SetPassCancelBtn">Cancel</string>
    <string name="syncAlertCancelStr">Cancel</string>
    <string name="syncCancelButStr">Cancel</string>
    <string name="syncEmailSyncOrRestoreDialogCancelButtonStr">Cancel</string>
    <string name="WIFINotConnectedDialogCancelBtnStr">Cancel</string>
    <string name="cancelEditStr">Cancel Edit Type</string>
    <string name="ProcessCancelledStr">Canceling operation, please wait ...</string>
    <string name="fontSizeStr">Change Font Size</string>
    <string name="changePasswordSummaryStr">Change or clear the password used in mSecure</string>
    <string name="changePasswordStr">Change Password</string>
    <string name="CheckingDropBoxExist">Checking for Dropbox sync file</string>
    <string name="recommendEmailSubjectStr">Checkout mSecure Password Manager</string>
    <string name="recommendBodyStr">Checkout mSecure Password Manager for Android, iOS, Windows and MacOS\nhttp://msevensoftware.com</string>
```

✓ URL:

Search for text:

http ☑ Auto search

Search definitions of: ☐ Class ☐ Method ☐ Field ☐ Code ☑ Resource ☐ Comments

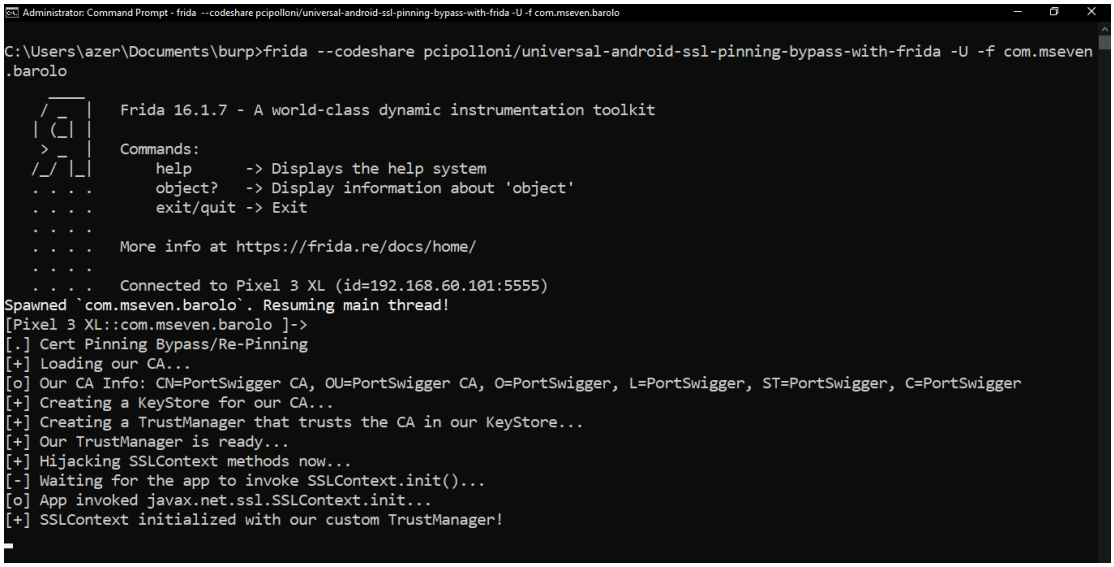
Search options: ☐ Case-insensitive ☐ Regex ☐ Active tab only

```xml
<public type="layout" name="http_authentication" id="0x7f030038" />
<string name="recommendBodyStr">欢迎了解用于Android、iOS、Windows和MacOS系统的mSecure密码助手。\nhttp://msevensoftware.com</string>
<string name="recommendBodyStr">歡迎瞭解用於Android、iOS、Windows和MacOS系統的mSecure密碼助手。\nhttp://msevensoftware.com</string>
<string name="recommendBodyStr">Android, iOS, Windows □□□ MacOS□ □□ mSecure □□□□ □□□□ □□□□ \nhttp://msevensoftware.com</string>
<string name="recommendBodyStr">Android、iOS、Windows、そしてMac用のmSecureパスワードマネージャーを確認してください。\nhttp://msevensoftware.com<
<string name="recommendBodyStr">Оформить заказ на Менеджер Паролей mSecure для Android, iOS, Windows и MacOS\nhttp://msevensoftware.com</st
<string name="recommendBodyStr">Verifique pelo gerenciador de senhas mSecure para Android, iOS, Windows e MacOS\nhttp://msevensoftware.com<
<string name="recommendBodyStr">Verifique el administrador de contraseñas de mSecure de Android, iOS, Windows y MacOS\nhttp://msevensoftwar
<string name="recommendBodyStr">Kasa mSecure Password Manager dla Androida, iOS, Windowsa i MacOS\nhttp://msevensoftware.com</string>
<string name="recommendBodyStr">Verifica Gestore password di mSecure per Android, iOS, Windows e MacOS\nhttp://msevensoftware.com</string>
<string name="recommendBodyStr">Commandez le Gestionnaire de mot de passe mSecure pour Android, iOS, Windows et MacOS\nhttp://msevensoftwar
<string name="about_privacy_web_str">http://msevensoftware.com/privacy</string>
<string name="about_permissions_web_str">http://msevensoftware.com/android_users_guide#Permissions</string>
<string name="about_help_web_str">http://msevensoftware.com/android_users_guide</string>
<string name="about_privacy_web_str">http://msevensoftware.com/privacy</string>
<string name="about_permissions_web_str">http://msevensoftware.com/android_users_guide#Permissions</string>
<string name="about_help_web_str">http://msevensoftware.com/android_users_guide</string>
<string name="about_privacy_web_str">http://msevensoftware.com/privacy</string>
<string name="about_permissions_web_str">http://msevensoftware.com/android_users_guide#Permissions</string>
<string name="about_help_web_str">http://msevensoftware.com/android_users_guide</string>
<string name="about_privacy_web_str">http://msevensoftware.com/privacy</string>
```

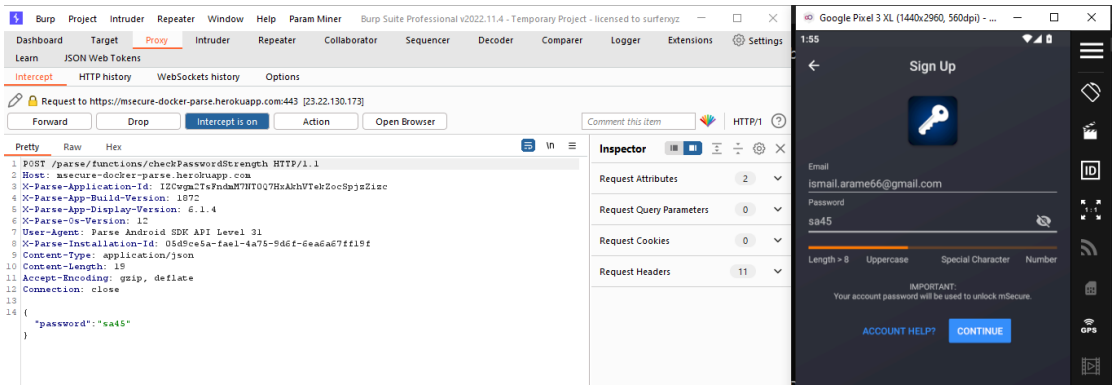
Load all | Load more | Stop | Found 50+ | Sort results

Dynamique Analysis :

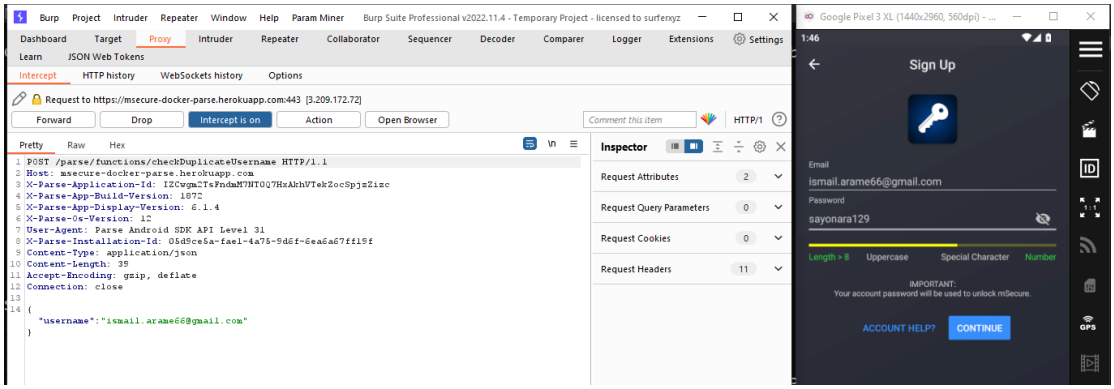
- ✓ On lance ssl pinning certificate bypass comme l'exemple precedant mains maintenant pour l'application msecure



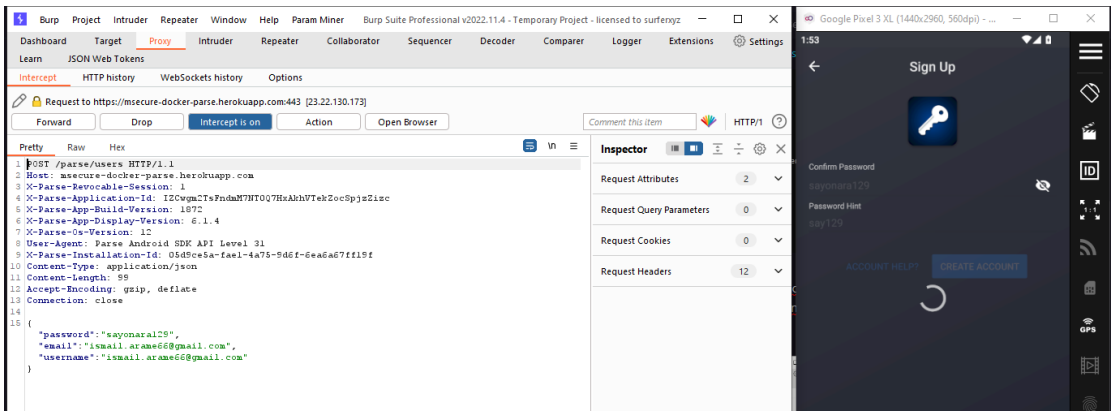
- ✓ J'ai commencé après par créer un compte sur mSecure, l'application fait un check du password simultanément en envoyant une requette checkPasswordStrength vers le server mSecure :



- ✓ aprés l’application fait un check pour vérifier la duplication de email c.a.d s'il est déjà enregistré et s'il est unique en utilisant la fonction checkDuplicateUsername

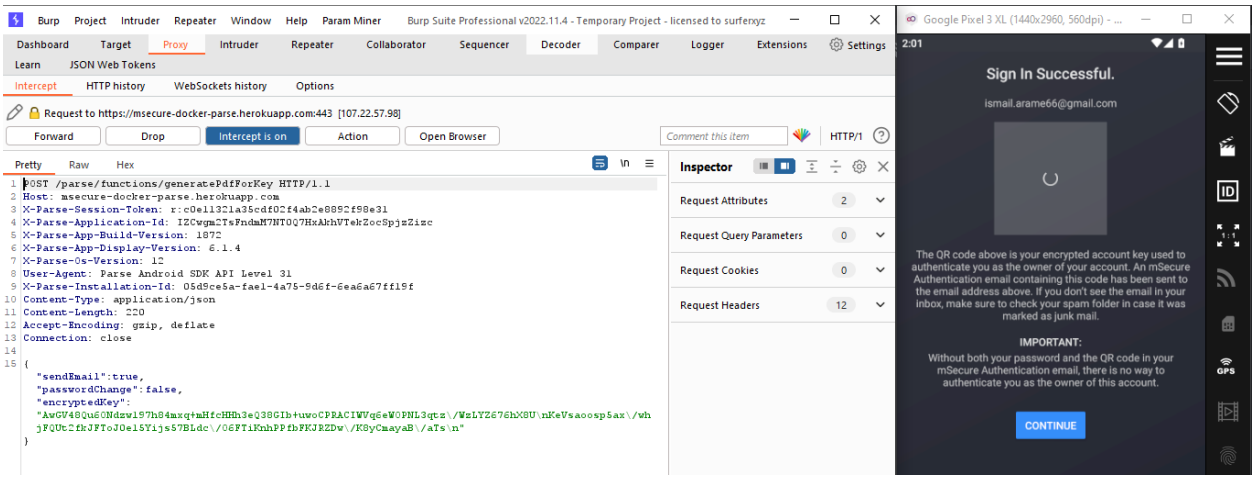


- ✓ apres la confirmation du password on remarque que l'email et le password et intercepete par burpsuite en text clair, d’où un attacker qui est entrain d’espionner va avoir l’access à mon email et password

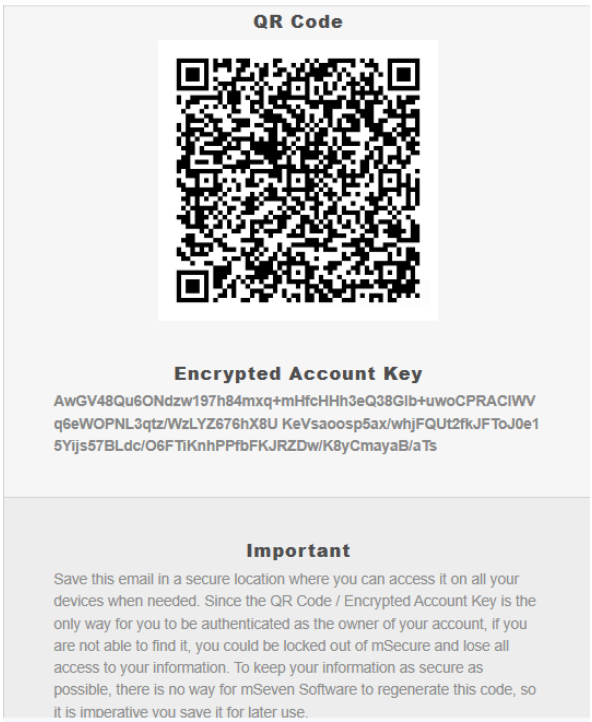


- ✓ D’après mSecure, lors de mon premier login, mSecure génére un super secret password appelé ‘’Account Key’’ et il l’utilise pour chiffrer un text clair connu et ils le stocke sur mon compte

✓ dans leurs system. Après mon mot de passe personnel est utlisé parsuite pour chiffre l'account key, et enfin ils m'envoient un émail contenant l'account key chiffré qui est nécessaire pour m'authentifier en tant que propriétaire de mon compte:



✓ l'email que j'ai recu.



➢ Bitwarden :

Static analysis :

La premiere étape est de pull l'apk au disque externe de notre pc.

```
PS C:\Users\PC\Downloads\Hello\Hello> adb pull /data/app/~~U7lzYHquPHOl7TGQ9DE8Ww==/com.x8bit.bitwarden-Wr7d2DeUQ0RdK85Md8HCVw==/base.apk bitwarden.apk
/data/app/~~U7lzYHquPHOl7TGQ9DE8Ww==/com.x8bit.bitwarden-Wr7d2DeUQ0RdK85Md8HCVw==/base.apk: 1 file pulled, 0 skipped. 127.4 MB/s (26030332 bytes in 0.195s)
PS C:\Users\PC\Downloads\Hello\Hello>
```

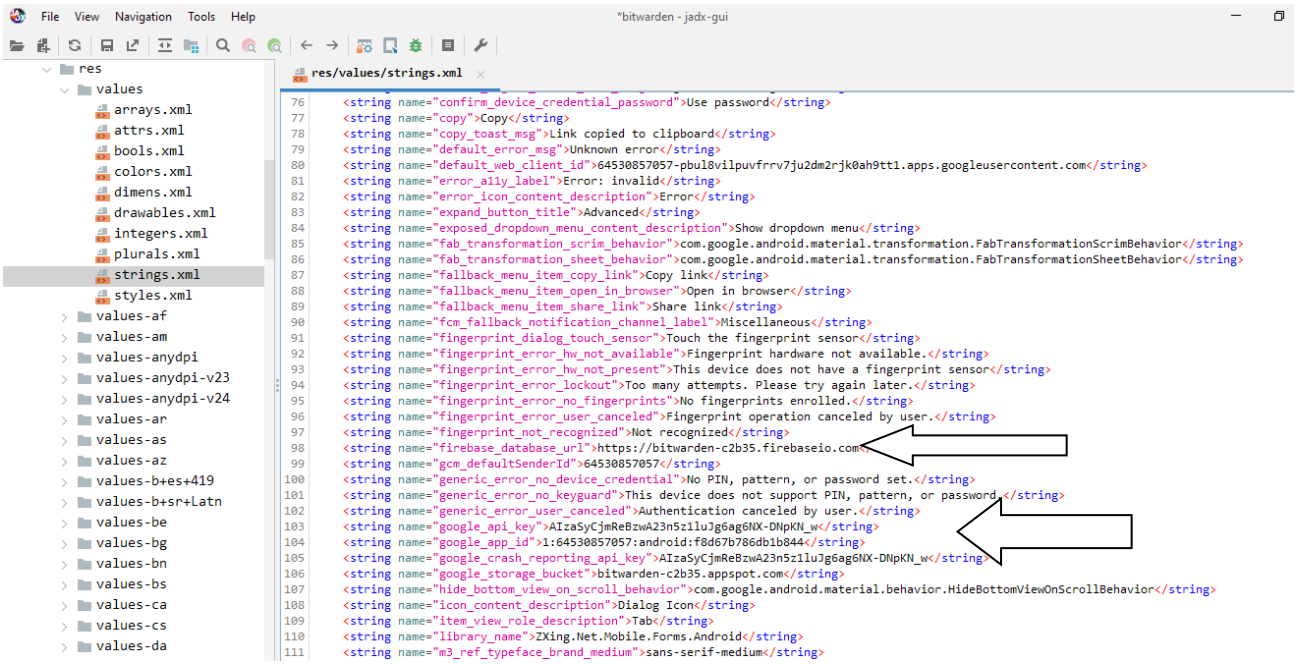
Apres la récuperation du l'apk , on utilise jadx pour la décompilation et obtenir le code source et les fichers de l'apk ,On commence par l'analyse du fichier Android.Manifest.XML

AndroidManifest.xml | firebase-components.properties | resources.arsc | res/values/strings.xml | res/values/styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="8588" android:versionName="2023.12.0" android:installLocat
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="33"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.NFC"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
    <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="com.samsung.android.providers.context.permission.WRITE_USE_APP_FEATURE_SURVEY"/>
    <uses-feature android:name="android.hardware.camera" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <queries>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <data android:scheme="http"/>
        </intent>
        <intent>
            <action android:name="android.intent.action.VIEW"/>
            <data android:scheme="https"/>
        </intent>
    </queries>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <permission android:name="com.x8bit.bitwarden.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION" android:protectionLevel="signature"/>
    <uses-permission android:name="com.x8bit.bitwarden.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <application android:theme="@style/LaunchTheme" android:label="Bitwarden" android:icon="@mipmap/ic_launcher" android:name="com.x8bit.bitwarden.M
        <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.x8bit.bitwarden.fileprovider"
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
        </provider>
        <meta-data android:name="android.max_aspect" android:value="2.1"/>
        <meta-data android:name="android.content.APP_RESTRICTIONS" android:resource="@xml/app_restrictions"/>
        <meta-data android:name="com.samsung.android.sdk.multiwindow.enable" android:value="true"/>
        <meta-data android:name="com.samsung.android.sdk.multiwindow.penwindow.enable" android:value="true"/>
        <meta-data android:name="com.lge.support.SPLIT_WINDOW" android:value="true"/>
```

✓ Res/values/strings.xml:

On trouve deux API_KEYs et un lien d'une database qui peut être usable pour decouvrir une vulnerabilité.



✓ Analyse du code source :

```
    public static class Pre28Implementation implements SignaturesCompat {
        Pre28Implementation() {
        }

        @Override // androidx.browser.trusted.PackageIdentityUtils.SignaturesCompat
        public List<byte[]> getFingerprintsForPackage(String name, PackageManager pm) throws PackageManager.NameNotFoundException
            PackageInfo packageInfo = pm.getPackageInfo(name, 64);
            ArrayList arrayList = new ArrayList(packageInfo.signatures.length);
            for (Signature signature : packageInfo.signatures) {
                byte[] certificateSHA256Fingerprint = PackageIdentityUtils.getCertificateSHA256Fingerprint(signature);
                if (certificateSHA256Fingerprint == null) {
                    return null;
                }
                arrayList.add(certificateSHA256Fingerprint);
            }
            return arrayList;
        }

        @Override // androidx.browser.trusted.PackageIdentityUtils.SignaturesCompat
        public boolean packageMatchesToken(String name, PackageManager pm, TokenContents token) throws IOException, PackageManage
            List<byte[]> fingerprintsForPackage;
            if (name.equals(token.getPackageName()) && (fingerprintsForPackage = getFingerprintsForPackage(name, pm)) != null) {
                return token.equals(TokenContents.create(name, fingerprintsForPackage));
            }
            return false;
        }
    }

    static byte[] getCertificateSHA256Fingerprint(Signature signature) {
        try {
            return MessageDigest.getInstance("SHA256").digest(signature.toByteArray());
        } catch (NoSuchAlgorithmException unused) {
            return null;
        }
    }
}
```

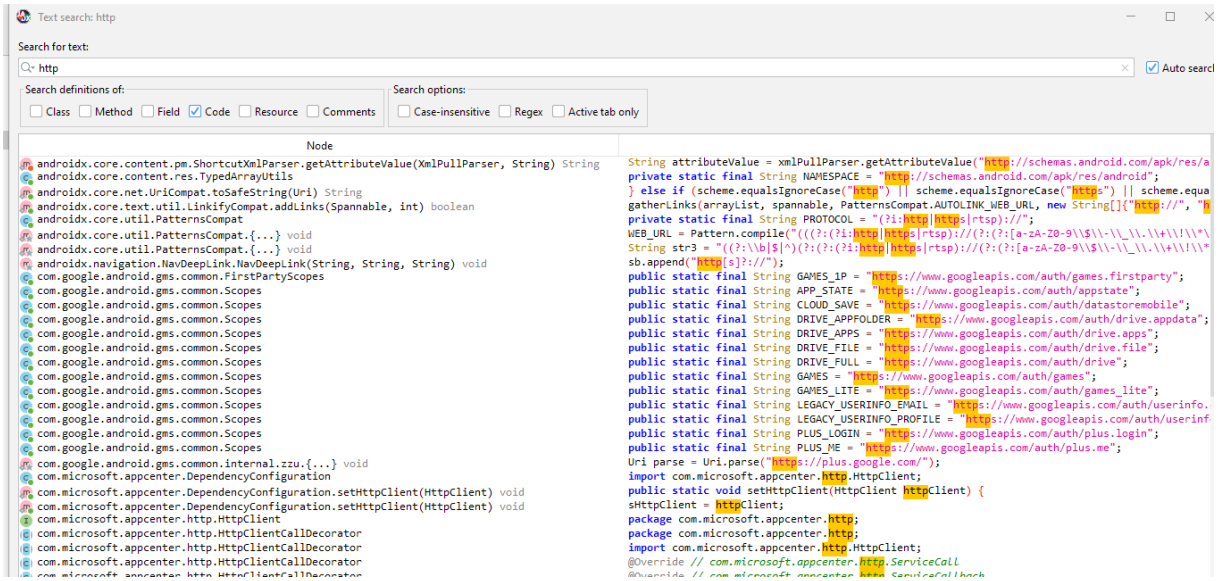
```
/* loaded from: classes.dex */
class CryptoAesHandler implements CryptoHandler {
    @Override // com.microsoft.appcenter.utils.crypto.CryptoHandler
    public String getAlgorithm() {
        return "AES/CBC/PKCS7Padding/256";
    }

    @Override // com.microsoft.appcenter.utils.crypto.CryptoHandler
    public void generateKey(CryptoUtils.ICryptoFactory cryptoFactory, String alias, Context context) throws Exception {
        Calendar calendar = Calendar.getInstance();
        calendar.add(1, 1);
        CryptoUtils.IKeyGenerator keyGenerator = cryptoFactory.getKeyGenerator("AES", "AndroidKeyStore");
        keyGenerator.init(new KeyGenParameterSpec.Builder(alias, 3).setBlockModes("CBC").setEncryptionPaddings("PKCS7Padding").setKeySize(256)
        keyGenerator.generateKey();
    }

    @Override // com.microsoft.appcenter.utils.crypto.CryptoHandler
    public byte[] encrypt(CryptoUtils.ICryptoFactory cryptoFactory, int apiLevel, KeyStore.Entry keyStoreEntry, byte[] input) throws Exceptior
        CryptoUtils.ICipher cipher = cryptoFactory.getCipher("AES/CBC/PKCS7Padding", "AndroidKeyStoreBCWorkaround");
        cipher.init(1, ((KeyStore.SecretKeyEntry) keyStoreEntry).getSecretKey());
        byte[] iv = cipher.getIV();
        byte[] doFinal = cipher.doFinal(input);
        byte[] bArr = new byte[iv.length + doFinal.length];
        System.arraycopy(iv, 0, bArr, 0, iv.length);
        System.arraycopy(doFinal, 0, bArr, iv.length, doFinal.length);
        return bArr;
```

✓ **Pour generer le cle de chiffrement Bitwarden utilise AES CBC mode et SHA256 Pour la signature de certificate.**

✓ Keystore :

```
218             return new FingerprintManagerCompat.CryptoObject(mac);
                }
222             if (Build.VERSION.SDK_INT >= 30 && cryptoObject.getIdentityCredential() != null) {
224                 Log.e(TAG, "Identity credential is not supported by FingerprintManager.");
                }
                return null;
            }

            /* JADX INFO: Access modifiers changed from: package-private */
241         public static BiometricPrompt.CryptoObject createFakeCryptoObject() {
                try {
242                 KeyStore keyStore = KeyStore.getInstance(KEYSTORE_INSTANCE);
243                 keyStore.load(null);
246                 KeyGenParameterSpec.Builder createKeyGenParameterSpecBuilder = Api23Impl.createKeyGenParameterSpecBuilder(FAKE_KEY_NAME, 3);
249                 Api23Impl.setBlockModeCBC(createKeyGenParameterSpecBuilder);
250                 Api23Impl.setEncryptionPaddingPKCS7(createKeyGenParameterSpecBuilder);
253                 KeyGenerator keyGenerator = KeyGenerator.getInstance("AES", KEYSTORE_INSTANCE);
255                 Api23Impl.initKeyGenerator(keyGenerator, Api23Impl.buildKeyGenParameterSpec(createKeyGenParameterSpecBuilder));
256                 keyGenerator.generateKey();
260                 Cipher cipher = Cipher.getInstance("AES/CBC/PKCS7Padding");
263                 cipher.init(1, (SecretKey) keyStore.getKey(FAKE_KEY_NAME, null));
265                 return new BiometricPrompt.CryptoObject(cipher);
                } catch (IOException | InvalidAlgorithmParameterException | InvalidKeyException | KeyStoreException | NoSuchAlgorithmException | NoSuchProvi
269                 Log.w(TAG, "Failed to create fake crypto object.", e);
                    return null;
                }
            }
```

✓ URLS :



```
Text search: http

Search for text:
http                                                                                    Auto search

Search definitions of:                                  Search options:
Class  Method  Field  Code  Resource  Comments          Case-insensitive  Regex  Active tab only

Node
androidx.core.content.pm.ShortcutXmlParser.getAttributeValue(XmlPullParser, String) String
androidx.core.content.res.TypedArrayUtils
androidx.core.net.UriCompat.toSafeString(Uri) String
androidx.core.text.util.LinkifyCompat.addLinks(Spannable, int) boolean
androidx.core.util.PatternsCompat
androidx.core.util.PatternsCompat.{...} void
androidx.core.util.PatternsCompat.{...} void
androidx.navigation.NavDeepLink.NavDeepLink(String, String, String) void
com.google.android.gms.common.FirstPartyScopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.Scopes
com.google.android.gms.common.internal.zzu.{...} void
com.microsoft.appcenter.DependencyConfiguration
com.microsoft.appcenter.DependencyConfiguration.setHttpClient(HttpClient) void
com.microsoft.appcenter.DependencyConfiguration.setHttpClient(HttpClient) void
com.microsoft.appcenter.http.HttpClient
com.microsoft.appcenter.http.HttpClientCallDecorator
com.microsoft.appcenter.http.HttpClientCallDecorator
com.microsoft.appcenter.http.HttpClientCallDecorator

String attributeValue = xmlPullParser.getAttributeValue("http://schemas.android.com/apk/res/a
private static final String NAMESPACE = "http://schemas.android.com/apk/res/android";
} else if (scheme.equalsIgnoreCase("http") || scheme.equalsIgnoreCase("https") || scheme.equa
gatherLinks(arrayList, spannable, PatternsCompat.AUTOLINK_WEB_URL, new String[]{"http://", "h
private static final String PROTOCOL = "(?i:http|https|rtsp)://";
WEB_URL = Pattern.compile("(((?:(?i:http|https|rtsp)://(?:(?:[a-zA-Z0-9\\$\\-\\_\\.\\+\\!\\*\\
String str3 = "((?:\\b|$|^)(?:(?:(?i:http|https|rtsp)://(?:(?:[a-zA-Z0-9\\$\\-\\_\\.\\+\\!\\*
sb.append("http[s]?://");
public static final String GAMES_1P = "https://www.googleapis.com/auth/games.firstparty";
public static final String APP_STATE = "https://www.googleapis.com/auth/appstate";
public static final String CLOUD_SAVE = "https://www.googleapis.com/auth/datastoremobile";
public static final String DRIVE_APPFOLDER = "https://www.googleapis.com/auth/drive.appdata";
public static final String DRIVE_APPS = "https://www.googleapis.com/auth/drive.apps";
public static final String DRIVE_FILE = "https://www.googleapis.com/auth/drive.file";
public static final String DRIVE_FULL = "https://www.googleapis.com/auth/drive";
public static final String GAMES = "https://www.googleapis.com/auth/games";
public static final String GAMES_LITE = "https://www.googleapis.com/auth/games_lite";
public static final String LEGACY_USERINFO_EMAIL = "https://www.googleapis.com/auth/userinfo.
public static final String LEGACY_USERINFO_PROFILE = "https://www.googleapis.com/auth/userinf
public static final String PLUS_LOGIN = "https://www.googleapis.com/auth/plus.login";
public static final String PLUS_ME = "https://www.googleapis.com/auth/plus.me";
Uri parse = Uri.parse("https://plus.google.com/");
import com.microsoft.appcenter.http.HttpClient;
public static void setHttpClient(HttpClient httpClient) {
sHttpClient = httpClient;
package com.microsoft.appcenter.http;
package com.microsoft.appcenter.http;
import com.microsoft.appcenter.http.HttpClient;
@Override // com.microsoft.appcenter.http.ServiceCall
```

Dynamic analysis :

- ✓ **Push frida server in our device virtuel**

```
C:\Users\PC\Downloads>adb push "C:\Users\PC\Downloads\PassManagers\frida-server-16.1.10-android-x86" data/local/tmp
C:\Users\PC\Downloads\PassManagers\frida-server-16.1.10-an...le pushed, 0 skipped. 81.2 MB/s (51684860 bytes in 0.607s)

C:\Users\PC\Downloads>
```

- ✓ **Turn frida server to an executable and exe it and connect to frida server.**

```
130|generic_x86:/ $ cd data/local/tmp
generic_x86:/data/local/tmp $ ls
frida-server-16.1.10-android-x86  oat
generic_x86:/data/local/tmp $ chmod 777 frida-server-16.1.10-android-x86
generic_x86:/data/local/tmp $
```

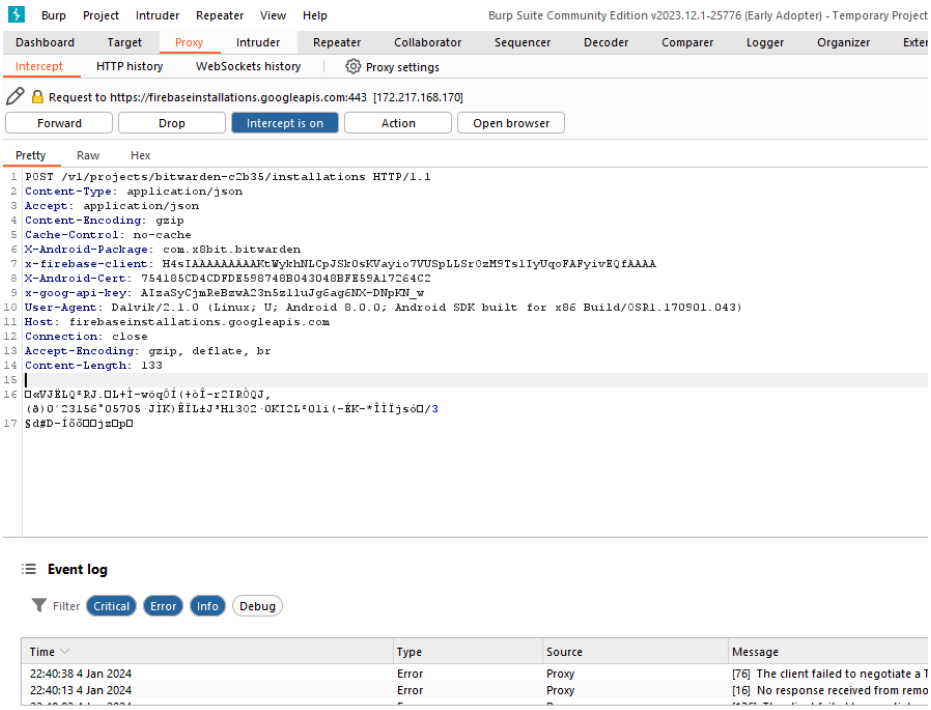
```
27|generic_x86:/data/local/tmp # ls
rida-server-16.1.10-android-x86  oat
eneric_x86:/data/local/tmp # ./frida-server-16.1.10-android-x86
```

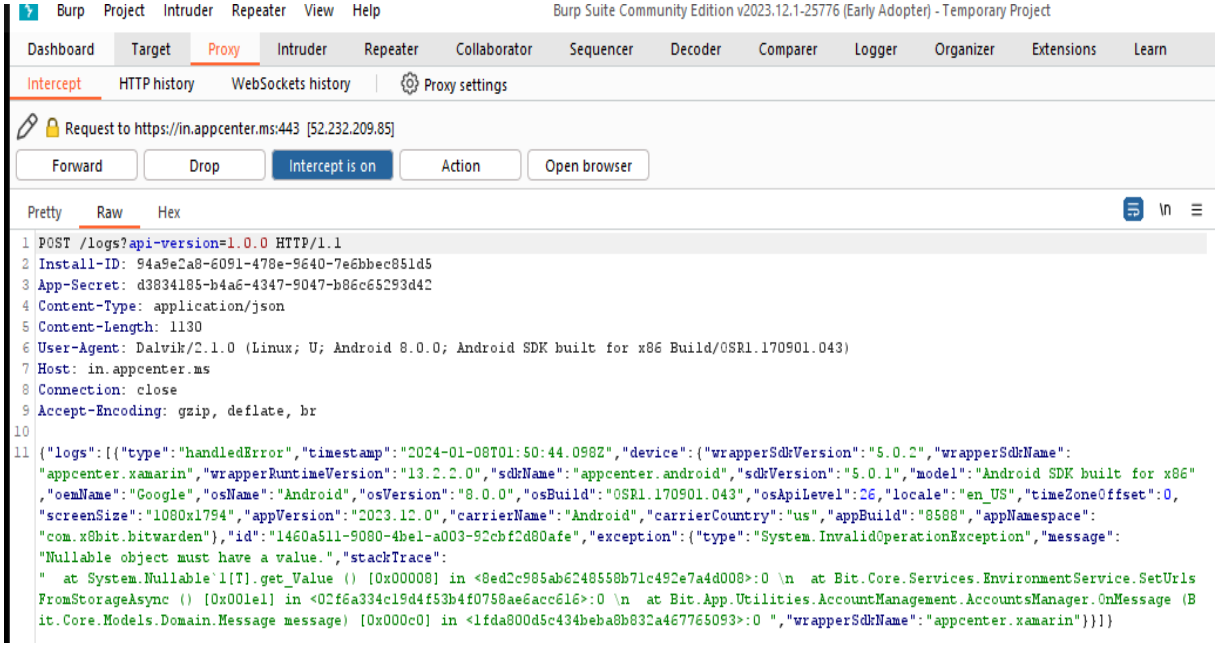
```
:\Users\PC>frida-ps -U
PID  Name
---  ----------------------------------------------------
040  Chrome
358  Gmail
129  Google
766  Google Play Store
484  Messages
554  Phone
951  Photos
011  YouTube
388  adbd
394  android.hardware.audio@2.0-service
043  android.hardware.biometrics.fingerprint@2.1-service
395  android.hardware.camera.provider@2.4-service
396  android.hardware.configstore@1.0-service
397  android.hardware.drm@1.0-service
398  android.hardware.drm@1.0-service.widevine
399  android.hardware.gatekeeper@1.0-service
031  android.hardware.gnss@1.0-service
400  android.hardware.graphics.allocator@2.0-service
401  android.hardware.graphics.composer@2.1-service
337  android.hardware.keymaster@3.0-service
032  android.hardware.media.omx@1.0-service
402  android.hardware.power@1.0-service
403  android.hardware.sensors@1.0-service
404  android.hardware.wifi@1.0-service
393  android.hidl.allocator@1.0-service
188  android.process.acore
895  android.process.media
019  audioserver
020  cameraserver
764  com.android.chrome
397  com.android.phone
825  com.android.printspooler
295  com.android.systemui
250  com.android.vending
```

- ✓ **Apres on va faire le même processus qu'on a déjà fait dans les applications precédent, créer un listener utilisant Burpsuite et configurer le proxy dans notre device and push the ca certificate.**
- ✓ **Apres fair tous ca on va bypass SSL pinning using frida.**

```
:\Users\Document\AppData\Roaming\Python\Python39\Scripts>frida --codeshare pcipolloni/universal-android-ssl-pinning-byp
ss-with-frida -U -f com.x8bit.bitwarden

   ____
  / _  |   Frida 16.0.8 - A world-class dynamic instrumentation toolkit
 | (_| |
  > _  |   Commands:
 /_/ |_|       help      -> Displays the help system
 . . . .       object?   -> Display information about 'object'
 . . . .       exit/quit -> Exit
 . . . .
 . . . .   More info at https://frida.re/docs/home/
 . . . .
 . . . .   Connected to Nexus 4 (id=192.168.72.101:5555)
pawned `com.x8bit.bitwarden`. Resuming main thread!
Nexus 4::com.x8bit.bitwarden ]->
.] Cert Pinning Bypass/Re-Pinning
+] Loading our CA...
o] Our CA Info: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
+] Creating a KeyStore for our CA...
+] Creating a TrustManager that trusts the CA in our KeyStore...
+] Our TrustManager is ready...
+] Hijacking SSLContext methods now...
-] Waiting for the app to invoke SSLContext.init()...
```

- ✓ **On a réussi de faire bypass SSL pinning et voila la resultat ,on remarque le traffic .**





- ✓ **voila une requete http n'est pas chiffré ,on peut utilisé ca pour être MITM.**