Mobility as a Service(MaaS) and Privacy

An analysis of the Security and Privacy of MaaS mobile applications

Master Security and Network Engineering
University of Amsterdam

Research Project 1
Project Report
Version: 1.1

Author:
Alexander Blaauwgeers
alexander.blaauwgeers@os3.nl

Supervisor:
Alex Stavroulakis
stavroulakis.alex@kpmg.nl

November 26, 2019

Abstract

This research investigated the security and privacy of Mobility-as-a-Service(MaaS) Android applications. The variation within the MaaS applications is large. In this research we focused on a small amount of applications. Te results of this research show that personally identifying information (PII), i.e. Mobile Carrier, Wireless SSID's and installation of applications of competitors, are used within the MaaS applications. We also have found reused One Time Passwords(OTP), within one of the applications.

Acknowledgements

I would like to thank my supervisor Alex Stavroulakis, and Vincent van Dongen for their support during this project.

Contents

1 Introduction

The way how people are traveling has changed in the recent past. They began doing their travel research by using smart mobile devices. As times changed, a growing number of new technologies and innovative sharing services appeared, combined with disruptive trends. This fundamentally reformed the public opinion about owning means of transportation and what the interpretation of mobility fundamentally should be. [1]

These developments are accelerated by autonomous vehicles, journey and car sharing, but also smart infrastructure like 'The Internet of Things' (IoT) and contactless smart card payment systems have far-reaching consequences for our existing transport system, namely faster, cheaper, cleaner and safer mobility. Mobility-as-a-Service(MaaS) stands for a mobility concept, in which the consumer uses different means of transport, on pay by usage basis, as part of the Smart City. MaaS is often compared or associated with Spotify and Netflix, as the 'Netflix' of mobility.[1][2][3]

As usage increases, also the questions around digital security and privacy arise, because the data those applications contain might be valuable for businesses- and secret service intelligence.[4] In the past they had to keep track of movements with manual means. Nowadays, smartphone (applications) contain a lot more information about our daily life.[5].

1.1 Research question

The main question for this research is:

What type of personal information is collected by Mobility-as-a-Service (MaaS) applications, how is this data secured and is this data necessary to operate the service offered to the user?

The research question can be divided into multiple sub-questions:

1. What kind of MaaS applications are available and what service do they offer to the user?
2. What techniques are used to securely send personal information? And how can these techniques be bypassed?
3. What kind of personal information is collected and send the MaaS applications by looking at their trafic and data storage?

2 Related work

This chapter briefly discusses the relevance of Mobility-as-a-Service (MaaS), which kind of MaaS applications are operated. We will also give a classification based on related work.

2.1 Relevance of privacy related to MaaS

Costantini[4] has written in his overview that the data of MaaS has such huge economic value, which makes it important to establish regulations and restrictions on if and how such information should be transferred or shared with other parties for commercial purposes. Furthermore, the data of MaaS might also contain Personal Identifiable Information (PII). This has been regulated mainly by General Data Protection Regulation (EU GDPR).

GDPR[6] provided companies specific criteria and rules which state that users (Data subjects) have the right to know what personal data companies store and process[7]. This includes the source of their personal data, the purpose of processing, and the length of time the data will be held, among other items. Most importantly, they have a right to be provided with the personal data of theirs that companies are processing.

2.2 Taxonomy of MaaS

Sochor[8] has written in her topological approach about the different viewpoints to classify MaaS applications. She wrote that you can differ them 1) By Service and by 2) By the level of Integration.

The **first** classification is related to **the service** they offer (i.e. transport services, rental, information, payment and ticketing). The Financial profit model is based on the bundling and organisational integration which entails e.g. a subscription to trips with different modes. This can be Integrated payment or invoicing for multiple private taxi companies, so for the customer they will look like one organisation. (e.g. Taxicompany "a" and "b" are both offering their services via TaxiApp "x".)

The **second** classification is related to **the level of integration** they offer. Where Level 0 means 'no intergration' and 4 means 'integration of societal goals'. Level 1 means 'integration of information' which include servies to exhange information rather then offering the service beneath it. Public transportation travel planners are an example of a Level 1 application.

1. Integration of information
2. Integration of booking and payment
3. Integration of the service offer
4. Integration of societal goals

2.3 State of the art

This notion of economic value and classification is important when you take a look at the privacy of different applications. Because the type of service and integration they offer also has a huge influence on the data they need to process. An 'Integration of information' (level 1) application for example is able to run entirely stateless. There is no need to create a profile or user account. When the level of integration increases, so does the amount of personal data needed. For example, if you integrate booking and payment(level 2) the application needs identifiable information and possibly a payment profile (e.g. creditcard or service specific deposit).

3 Methodology

This chapter discusses the kind of experiments needed in order to answer the research questions. It also defines the limitations of the research and the techniques used.

3.1 Project description

In this section we reflect on the original project description as defined in the proposal.

Goal: The goal of this project is to analyse the security and privacy of MaaS mobile applications since GDPR came in to effect.

Approach: During the project the MaaS Android applications will be installed to inspect the data sent. Possible techniques to protect the secure transfer of data against have to be bypassed by tools like a man-in-the-middle(MitM) proxy.

Categorisation of the Experiments To structure the experiments, the methodology defined by The Open Web Application Security Project(OWASP) has been used. The experiments are based on the OWASP Mobile Security Project.

Scope: Due to the limited time available the project is scoped by Android Applications. The project will focus on the application and the traffic generated by using the application. The project is scoped by a limited amount of applications.

Selection of the Applications: For the experiment we selected a small set of applications which are popular within Europe. The applications were initially selected to have a mix within the field, based on the most popular apps.

Several apps like Greenwheels[1] and Car2go[2] require an extensive registration procedure with validation of bankdetails and identification documents (eg. Driver's licence) have been skipped.

Uber[3] has been skipped because the application crashed during the experiment. It is important to note that Yandex.Taxi merged with Uber.[9].Yandex.Taxi is incorporated in the Netherlands under the name MLU B.V, also active within the EU.

The final longlist of applications which we checked; Yandex.Taxi[4], Taxibeat[5], TaxiBolt[6], NSapp[7], OVapi[8] and Lime[9].

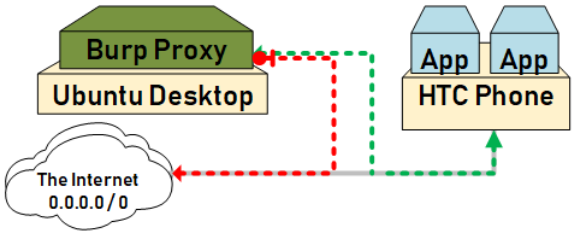

Figure 1: *Used Man in the Middle(MitM) network topology.*

[1] https://www.greenwheels.com
[2] https://www.car2go.com
[3] https://www.uber.com
[4] https://taxi.yandex.com
[5] https://thebeat.co
[6] https://bolt.eu
[7] https://www.ns.nl
[8] https://ovapi.nl
[9] https://www.li.me

3.2 Setup of the Experiments

In figure 1 the setup is shown. The Android applications will be installed on a Physical Device and/or Emulator. The certificate store of the devices has been modified which adds the ability to listen in to the SSL protected traffic by a man-in-the-middle(MitM) Proxy.

3.3 Techniques used to securely send data

During the experiments the applications where tested on a physical phone. This HTC10 device was running Android version 8. On this version of Android, by default, applications do not trust the user-added CA store. Trusted Root Certificates(TRC) on Android are stored in a System Store. The Android developer guide describes the new policy as follows: " By default, secure connections (using protocols like TLS and HTTPS) from all apps trust the pre-installed system CAs, and apps targeting Android 6.0 (API level 23) and lower also trust the user-added CA store by default. "[10] To be able to preform our experiment with the MitM proxy we added our CA certificate to the System store. For this reason we needed to have the used phone rooted.

3.4 Tools used for the Experiments

To conduct the experiment we used the following tools have been used;

SOFTWARE

$T1$: **Frida Framework**
Frida[11] is a framework, used by pen-testers, to inject your foreign code and scripts into black box processes. This framework is used to bypass SSL certificate pinning within some applications. (e.g. Uber and Lime.bike)

$T2$: **Android Debugger (adb)**
Android Debug Bridge(adb)[12][13] is a command-line tool that lets you communicate with an Android device for which it provides access to the Unix shell.

$T3$: **FakeGPS**
FakeGPS[14] is a Android tool to fake GPS location.

$T4$: **BurpSuite**
BurpSuite[15] is a Java based application used to test and analyse the security of applications. It is used as Man-in-the-Middle(MitM) proxy.

$T5$: **Google Play Store(Android App Market)**
The experiments have been conducted on the latest original version off the apps. Downloaded at 10 October 2019 from the Google Play store.

HARDWARE

$T5$: **Phone: HTC10** Running Android 8.0

$T6$: **Vodafone Mobile SIM**
A Dutch simcard to receive SMS text messages during the project. This card was not used before.

4 Results

This chapter discusses the results of the experiments. It briefly discusses the findings of personal data collected and send by the various applications.

4.1 Unintended Data Leakage

To offer the service to the users most MaaS applications need some form of location. By examining the traffic from the app to the backend we can see that most of them use a combination of 1) GPS location 2) Cellphone metadata 3) WiFi metadata as their source of location.

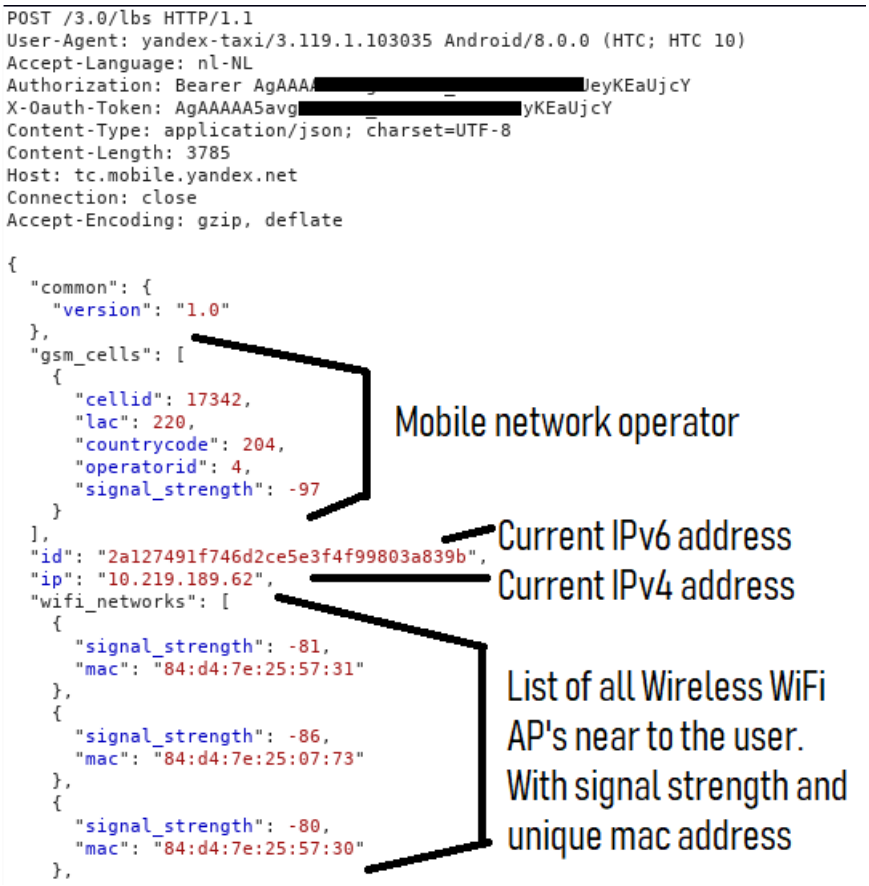

Figure 2: *Yandex: Cell and wireless network discovery.*

For example, when we look into figure 2, we see the traffic from Yandex.taxi to their back-end. In this traffic we can clearly identify what kind of information transferred. Even when the Global positioning system (GPS) function on the phone has been enabled and is fully functional. The application communicate the users Mobile operator.

In figure 2 we also see the operator `"204 04"` which is the ITU Mobile Network Code(MNC)[16] for `"VodafoneNL"`. Taxibeat uses next to this also the International Mobile Equipment Identity(IMEI) code as their Universally Unique Identifier (UUID) as seen in figure 3.

The same kind of behaviour has been spotted in some other applications. The results of TaxiBold have been added as appendix 7.1. In this results we also see that TaxiBold collects the Service Set Identifier(SSID), which known as the name of Wireless Networks. This method of location determination leads to the leakage of personal identifiable information.

4.2 Improper Platform Usage

Applications do have access to the list of other installed applications. Applications can compare this list, with a list of known competitor applications.

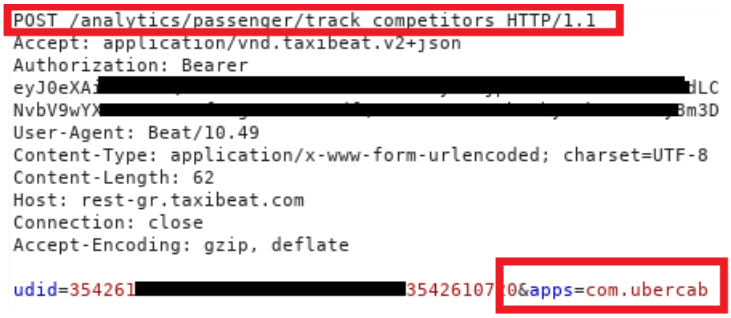

Figure 3: *Taxibeat: Competitor application discovery.*

If we take a look at figure 3, we see the traffic generated by the taxibeat-app. Within this traffic we see that the result[10] of this competitor application discovery has been reported to the back-end.

Based on this data the service has the ability, to give users likely to choose between competitive service operators, an extra discount, which is not available for other users. We have not found any concrete proof of abuse based on this advantage, so far. It is hard to prove within a small sample, because other factors, i.e. service demand, are also involved in the in the pricing of services.

4.3 Insecure Authentication

Most applications use some kind of Authentication. The most applications use a form of phonenumber verification over SMS method of Authentication. This type of Authentication, without other checks, is not always as secure as you would expect. Mulliner[17] wrote this in their Analysis of Weaknesses and Attacks SMS-OTP. Some of the tested applications (i.e. Taxibeat) only use SMS as their source of Authentication.
Based on the traffic in figure 4 we see that Taxibeat uses OAuth2 to secure their application. The OAuth2 recommendation[18] describes the following risk;

> **10.10. Credentials-Guessing Attacks** The authorization server MUST prevent attackers from guessing access tokens, authorization codes, refresh tokens, resource owner passwords, and client credentials."- RFC6749[18]

4.3.1 Example of a credential guessing attack

This means that a malicious user must be unable to tryout all the keyspace by guessing or a brute force method.

If we take a look at the **client credentials** of the Taxibeat application as shown in figure 4, they are limited to a Username and a Password. As username they use the phonenumber of the user and as password a SMS code received, see figure 5. Taxibeat only

[10] com.uber is the application name of Uber Taxi

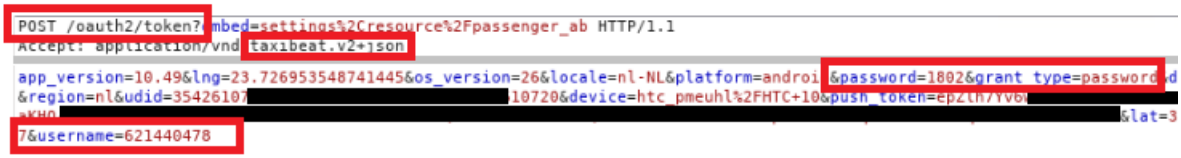

Figure 4: *Taxibeat authentication token.*

uses SMS to send this authorization code. Important to note is that when you reinstall the application on the same, or a different phone, the application, by default, does not request a new authorization code.

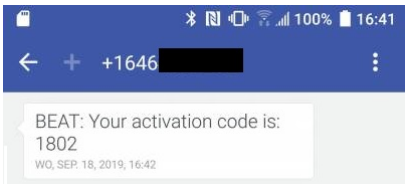

Figure 5: *Taxibeat sms with authentication code.*

Credential guessing attack Script We assumed the credentials are easy guessable. Due to the fact that there are only 10000 (10^4) possibilities, the code can be easily brute-forced. To prove this we tested this with following example script, shown in Listing 1.

```
#!/bin/bash
USERNAME="623456789" #correspond with a valid dutch phone number
for i in {0..9999..1}
  do
     echo "=========[ "+$i+" ]=========" >> output.log
     curl -d "app_version=10.49&lng=6.1371069&os_version=26&
   locale=nl-NL&platform=android&grant_type=password&
   device_density=5&region=nl&udid
   =354261072034234354261072034 23[REMOVED]&device=htc_pmeuhl%2
   FHTC10xxx&push_token=[REMOVED]&lat=52.2763029&username="+
   $USERNAME+"&password="+$i+"" -H "Accept: application/vnd.
   taxibeat.v2+json" -H "Authorization: Basic: [REMOVED]==" -X
   POST https://[REMOVED]auth2/token?embed=settings >> output.
   log
     echo "-----------------------" >> output.log
     sleep 10
done
```

Listing 1: Hijack session by guessing or brute-forcing code

Using a script we are able to loop over all the combinations within the 4 digit keyspace. However, to reduce the load on the server we limited this to 1700-1850, with a 10 second delay, which relates to 150 combinations. We assumed that we will got detected and blocked, but we were not catched. The output of the script in Listing 2.

The variables used in the request are;

- ▲ `username` => Phonenumber without the international part.
- ▲ `password` => A four digit code between 0000 and 9999.
- △ `grant_type` => Grant a auth2 token by using an password.
- △ `app_version` => Version 10.49 of the application
- △ `os_version` => Android API version (i.e. 26=8.0 27=8.1)
- △ `lat&lng` => Current location. This can be anywhere on earth, we also checked places where the service is not currently operated like Deventer(NL) and Funafuti(TV)
- △ `locale&region` => Used for internationalization of the phonenumber.
- △ `device` => Name of the device. Can be a random string of letters.
- △ `udid` => International Mobile Equipment Identity(IMEI) Unique number of the device. Can be anyting with exact 40 characters.
- △ `push&token` => session_id which can obtained on request.

```
=========[ +1800+ ]=========
{"errors":[-------------
=========[ +1801+ ]=========
{"errors":[{"message":"Your phone number and password
   combination was wrong","name":"_INVALID_CREDENTIALS_"}],"meta
   ":{"status":400,"version":"2","rtime":0.668,"host":"pe-247-
   hub-06"}}-----------------------
=========[ +1802+ ]=========
{"access_token":"eyJ0eXAiOiJKV1QiLCJhbGc...[REMOVED]...","
   token_type":"bearer","expires_in":14400,"scope":"passenger","
   settings":{"...[REMOVED]..."},"paypal":{"client_id":"AYzkhRD
   ...[REMOVED]...",}-------------
=========[ +1803+ ]=========
{"errors":[{"-------------
=========[ +1804+ ]=========
{"errors":[{"m-------------
```

Listing 2: snippet from the output log

In the response we received the same error on all replies between 1700 and 1850. Except for 1802 where we received a working token `access_token` accompanied by other configuration including `passenger_payment_details`. Therefor we can conclude that 1802 is the correct password of this user. We tried this password on a different phone and gained access to, in this case, our account.

4.4 Statistics and unidentified traffic

Almost all the applications are gathering statistics of the app usage. The contents of these statistics are comparable with `Google Analytics`. The software package `Crashlytics` is widely used. Yandex uses it's own tracking software named `Yandex metrica`.

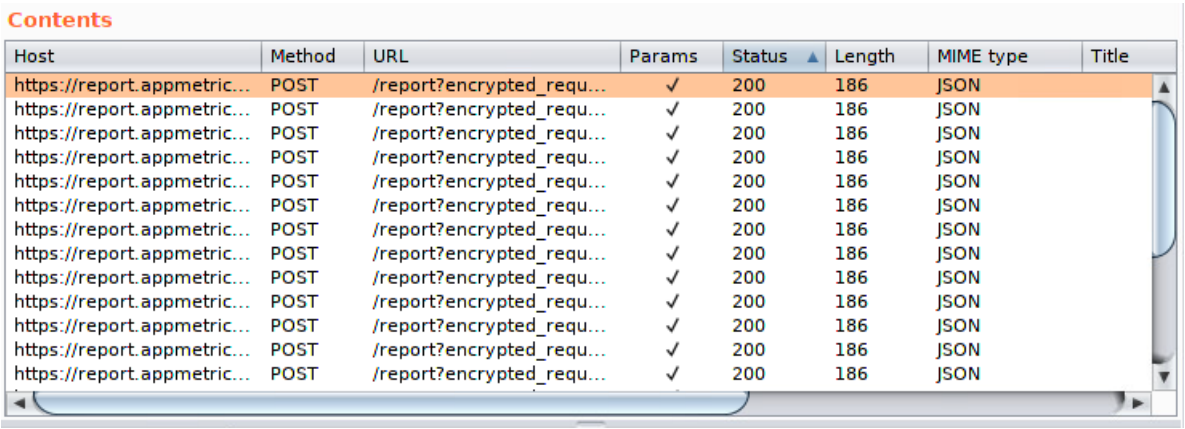

Figure 6: *Yandex report trafic overview.*

For the Yandex.Taxi application we observed some unidentified traffic, labeled with the `encrypted=true` parameter shown in Figure 6. We were unable to identify this traffic.

```
POST
/report?encrypted_request=1&deviceid=9f6e4e6fdfde4fe19dedeb5fe9aff171&uuid=b9184018791844bead8621ecc
c492c91&analytics_sdk_version_name=3.6.4&app_version_name=3.119.1&app_build_number=3103035&os_versio
n=8.0.0&os_api_level=26&analytics_sdk_build_number=45178&analytics_sdk_build_type=internal&app_debug
gable=0&locale=nl_NL&is_rooted=1&app_framework=native&attribution_id=1&api_key_128=67bb016b-be40-4c0
8-a190-96a3f3b503d3&app_id=ru.yandex.taxi&app_platform=android&model=HTC%2010&manufacturer=HTC&scree
n_width=2560&screen_height=1440&screen_dpi=640&scalefactor=4.0&device_type=phone&android_id=74b63584
64fd8f5b&adv_id=40bdaad6-3ae3-4463-9274-f9e3997d437b&limit_ad_tracking=0&request_id=2 HTTP/1.1
Accept: application/json
User-Agent: com.yandex.mobile.metrica.sdk/3.6.4.45178 (HTC 10; Android 8.0.0)
Send-Timestamp: 1571057413
Send-Timezone: 7200
Content-Type: application/x-www-form-urlencoded
Host: report.appmetrica.yandex.net
Connection: close
Accept-Encoding: gzip, deflate
Content-Length: 1456

□��R���z;Vi□□□C□□o□□□□'m□□E□□X□□;□□;□ON□□□s□□□G□□□Y□□□V□□□□C^□.Q1□□□□□□,□W□□k□□□□+3□□□□□□□
|X□f□□0*□□□□□i□□J□H□□&□□S□e;□'깝□□□□□□W□.z□□(□Mz□ə□□w□□"□□□□g□!c86□s□□~8□□{8SuQ>5
U□□km□□□□□□r□□□□□□□□Z]□□□gm□'□l□□□J□└K□u□□□-□}□b□□
□f□□□^}(□U□y□□□□? □□□□R□□□□□>:s
}kL□B□□□rM-□□gWb□□□G□□□□&1□□□□Q□?5□□5□O{t□-y□□□□,□□□6□D□□□D□□tB]□|R□B□□$□□□'□\lu'□□□□□□N□□□□x□8□□=
□1□□□.7□□p:□□□b□x)2□□□□r□□□`n□`j6□tI□□□D□□j□T/ '□□;a□□□□□O□□J+□□□□#d□@q3□`=B
```

Figure 7: *Yandex statistics report after stripping HTTPS.*

We preformed an entropy test on the blob of data transferred by Yandex.taxi, as shown in figure 7. This test was preformed using the tool `CyberChef`[11] published by the British Secret services (GCHQ). The result of this Shannon entropy measure can be found in Figure 8. On the Shannon entropy scale where the value 0 represents no randomness (eg. "aaaaaaaaaa"). A text containing standard human languages usually falls somewhere

[11] CyberChef[19]: https://gchq.github.io/CyberChef

between 3.5 and 5. Properly encrypted or compressed data of a reasonable length should have an entropy of over 7.5. The data transfered by Yandex.taxi scored an Shannon entropy higher then 7.9. Which means that it is highly likely to be encrypted or compressed.

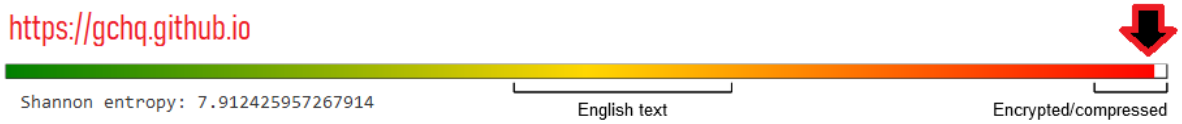

Figure 8: *Shannon entropy measure on Yandex.taxi blob.*

4.5 Summery of personal information

In this paragraph a summery of the personal information detected in the trafic of the MaaS application. This information is divided in two tables.

In table 1 the personal information entered by the user is shown. This is the minimal information which is needed to use the application. Some information is optional like a home address adress. Only the destination address is needed. This information is collected via user input. First and lastname are not verified.

In table 2 the personal information transferred to the backend server by the app is shown. This information is collected without user input. The information in table 2 might be incomplete, because some information is not detected within the traffic.

App	First and Lastname	Address	E-mail	Phonenumber	CreditCard
Yandex.Taxi	Y	N	N	Y	Y
TaxiBeat	Y	N	Y	Y	Y
TaxiBolt	Y	N	Y	Y	Y
NSapp	N	N	N	N	N
OVapi	N	N	N	N	N
Lime	Y	N	Y	Y	Y

Table 1: *Personal information minimally needed to use the basic functions of the app* ***Y****: Information Collected* ***N****: Not Collected or optional*

App	GPS	IP	Current WiFi	Other WiFi	Carrier	Root	Gyroscoop
Yandex.Taxi	Y	Y	Y	Y	Y	Y	X
Taxibeat	Y	Y	Y	X	X	X	X
TaxiBolt	Y	Y	Y	Y	Y	X	Y
NSapp	Y	Y	X	X	X	X	X
OVapi	Y	Y	X	X	X	X	X
Lime	Y	Y	X	X	X	X	X

Table 2: *Personal information transferred to the server by the app without user input* ***Y****: Information Collected* ***X****: Not detected in the traffic.*

5 Discussion

In this chapter we discuss the results. Some of the results are already briefly discussed within the previous chapters. We have to acknowledge some limitations in the results.

- During the experiment we only preformed a limited set of usecases. i.e. no real transactions have been preformed.
- We only preformed the experiments for a short period. (max. 30 min sessions)
- The experiment has been preformed only on a limited set of devices.
- The used devices had to be rooted before we were able to bypass the latest Android SSL security. This has no large impact since we did not aim to preform research in the authenticity of the data. Our research was about the traffic generated by the applications.
- Android applications may be aware of the fact that the phone has been rooted before usage. We see this in Yandex Trafic (figure 7).
- We did not modify the applications, we used the original latest version from the Google Play store.
- On the Taxibeat application we preformed a limited credential guessing attack. This attack was limited by the fact that did known the used phone number. In a realistic attack scenario this data may be gathered using i.e. social engineering. We also limited the keyspace to 1,5% of the total. But with this example we have proven it is feasible.
- We also admit that data encryption, as seen in the unidentified traffic, could have a legitimate purpose, i.e. to protect the personal identifiable information of the users.

6 Conclusion

In this chapter we answer the research question, by answering the sub questions.

This research investigated Mobility-as-a-Service(MaaS). MaaS stands for a mobility concept, in which the consumer uses different means of transport, on pay by usage basis. The variation within the MaaS applications is large. MaaS apps can be differentiated by `(1)Service` and by `(2)the level of Integration`.

For this research we selected a small set of applications which are popular within Europe. The applications were initially selected to have a mix within the field, based on the most popular apps.

The researched MaaS applications make use of the Android SSL library. Next to SSL most applications use some kind of SMS one-time password (OTP) to secure their users. Taxibeat uses this OTP multiple times, without limiting on the amount of attempts, which leads to insecure authentication. At least the amount of login attempts should be limited.

Some applications, e.g. TaxiBold, Yandex.Taxi, do collect personally identifiable information, i.e. Wireless SSID, Mobile Carrier. Furthermore, Taxibeat also registers applications of competitors like Uber, which is not needed for the service they offer.

6.1 Future work

Based on the results and we propose to the same research from from a experienced Law viewpoint. In this research we have taken a technical viewpoint, but we were unable to make a judgement on the minimal need of information for MaaS Applications. For this reason we assume that a GDPR Audit has relevance. We also propose to extend the research with other applications, other mobile platforms(i.e. iPhone) and web-only applications.

Besides, we propose to research GDPR compliance of Yandex.Metrika and the data gathered as part of this. Yandex.Metrika is used next to Google Analytics, to gain statistics of mobile application use. These applications are used within the EU region, especially the Eastern part of the Union.

References

[1] Durand at all. *Mobility-as-a-Service and changes in travel preferences and travel behaviour: a literature review*. KiM Netherlands Institute for Transport Policy Analysis. Sept. 17, 2018. URL: https://english.kimnet.nl/publications/documents-research-publications/2018/09/17/mobility-as-a-service-and-changes-in-travel-preferences-and-travel-behaviour-a-literature-review (visited on 06/17/2019).

[2] Tayyab Iqbal. *Urban Mobility: Role of taxation in the adaptation of Mobility as a Service (MaaS) and tool for Policy Making*. 2019. URL: https://www.theseus.fi/bitstream/handle/10024/172191/Tayyab_Iqbal.pdf.

[3] David König et al. *Technology for MaaS-Deliverable Nr 5*. 2017. URL: https://www.vtt.fi/sites/maasifie/PublishingImages/results/cedr_mobility_MAASiFiE_deliverable_5_revised_final.pdf.

[4] Federico Costantini. "MaaS and GDPR: an overview". In: *arXiv preprint arXiv:1711.02950* (2017).

[5] Katie Shilton. "Four billion little brothers? Privacy, mobile phones, and ubiquitous data collection". In: (2009).

[6] *EU2018 reform of EU data protection rules*. European Commission. May 25, 2018. URL: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf (visited on 09/23/2019).

[7] *Right of access by the data subject (art. 15 GDPR*. European Commission. May 25, 2018. URL: https://gdpr.eu/article-15-right-of-access/ (visited on 09/23/2019).

[8] Jana Sochor et al. "A topological approach to Mobility as a Service: A proposed tool for understanding requirements and effects, and for aiding the integration of societal goals". In: *Research in Transportation Business & Management* 27 (2018), pp. 3–14. ISSN: 22105395.

[9] *Yandex and Uber Complete the Combination of Their Ride-Sharing Businesses in Russia and Neighboring Countries*. Yandex. Feb. 7, 2018. URL: https://yandex.com/company/press_center/press_releases/2018/0207 (visited on 10/23/2019).

[10] Open Handset Alliance (ANDROID). *Network security configuration*. Mar. 2019. URL: https://developer.android.com/training/articles/security-config.html.

[11] Ole André Vadla Ravnås. *Frida Framework*. Sept. 2019. URL: https://www.frida.re/.

[12] Android. *Android Debug Bridge (adb)*. Mar. 2019. URL: https://developer.android.com/studio/command-line/adb.

[13] Hans-Christoph Steiner and Kai-Chung Yan. *Android Tools for Debian based systems*. Mar. 2019. URL: https://wiki.debian.org/AndroidTools.

[14] LexaApps. *FakeGPS*. June 2018. URL: https://play.google.com/store/apps/details?id=com.lexa.fakegps.

[15] Dafydd Stuttard. *Burp Suite*. Sept. 2018. URL: http://releases.portswigger.net/2018/09/professional-2006beta.html.

[16] ITU. *Mobile Network Codes (MNC) for the internationalidentification plan for public networks and subscriptions*. Dec. 15, 2018. URL: https://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212B-2018-PDF-E.pdf (visited on 11/11/2019).

[17] Collin Mulliner et al. “SMS-Based One-Time Passwords: Attacks and Defense”. In: *Detection of Intrusions and Malware, and Vulnerability Assessment*. Ed. by Konrad Rieck, Patrick Stewin, and Jean-Pierre Seifert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 150–159. ISBN: 978-3-642-39235-1.

[18] D. Hardt. *The OAuth 2.0 Authorization Framework*. RFC 6749. `http://www.rfc-editor.org/rfc/rfc6749.txt`. RFC Editor, Oct. 2012. URL: `http://www.rfc-editor.org/rfc/rfc6749.txt`.

[19] gchq. *CyberChef 9.11.0*. Nov. 6, 2019. URL: `https://github.com/gchq/CyberChef` (visited on 11/11/2019).

[20] Laura J. Nelson. *L.A. wants to track your scooter trips. Is it a dangerous precedent?* Los Angeles Times. Mar. 15, 2018. URL: `https://www.latimes.com/local/lanow/la-me-ln-los-angeles-scooter-surveillance-privacy-20190315-story.html` (visited on 09/23/2019).

[21] Erika Chin et al. “Measuring user confidence in smartphone security and privacy”. In: *Proceedings of the eighth symposium on usable privacy and security*. ACM. 2012, p. 1.

[22] OWASP Foundation. “OWASP Mobile Security Project top-10 2016”. In: *OWASP Foundation* (Feb. 16, 2017). URL: `https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10` (visited on 09/23/2019).

7 Appendix:

7.1 Additional Results Taxibold

Figure 9: *TaxiBold: Report of MobileCarrier inside statistics.*

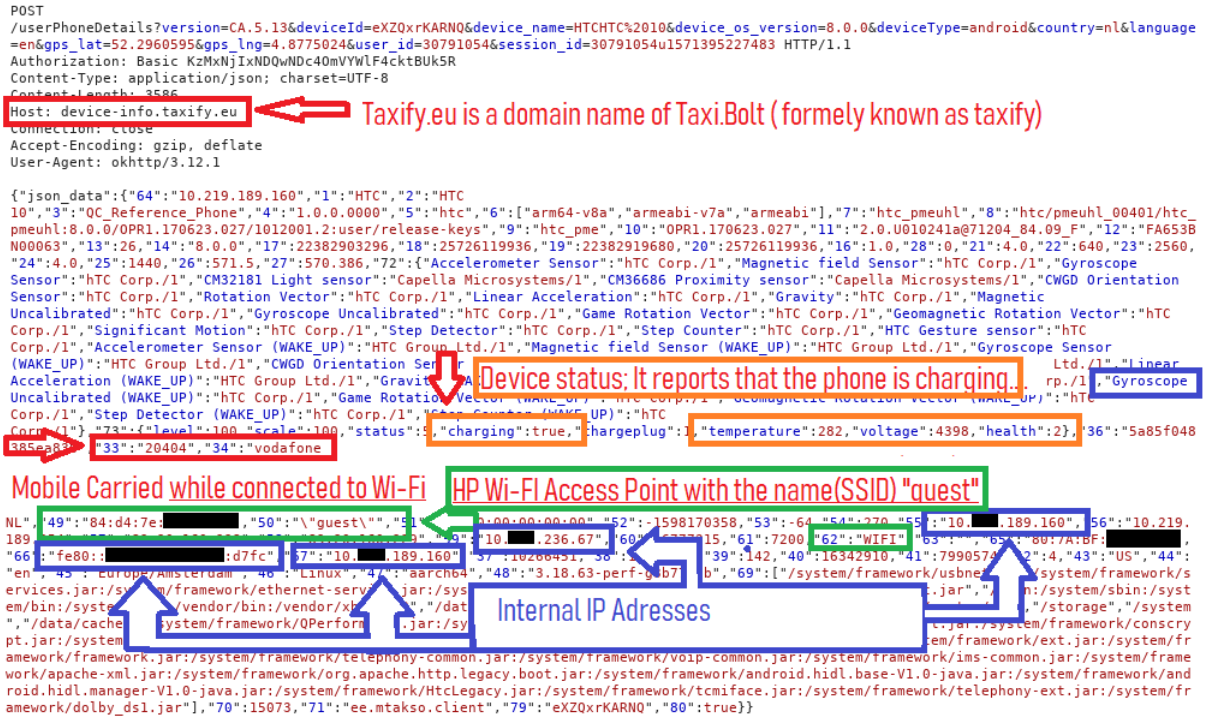

Figure 10: *TaxiBold: Full device report.*