



Plategate: Grab them by the plate!

After 6 years without an own car during my studies – which was a really hard time for me as a passionate car driver – I finally managed to put some money aside to get one and not just any car but THE number one nerd car – Tesla Model S. This car was definitely worth every year of car abstinence and made all 6 of them instantly pale into insignificance the minute I took a seat and hit the accelerator for the first time.

Nevertheless – business before pleasure – I first had to do a lot of paperwork regarding insurance and car registration. During that I stumbled across the following form (I live in Zurich, Switzerland, so unfortunately it is only available in German):

Gesuch um Sperrung der Halterdaten

Ich/wir beantrage/n, **alle auf meinen/unseren Namen im Kanton Zürich zugelassenen Kontrollschilder** mit einer Auskunftssperre zu belegen.

Name / Firma:

Vorname / Firmenzusatz:

Strasse / Nr.:

PLZ / Ort:

Telefon tagsüber:

E-Mail:

Bitte beachten Sie

Nach der Sperrung werden die Daten nur noch an Behörden weitergegeben, die Angaben von Amtes wegen benötigen. Über gesperrte Halter/innen wird an private Personen und Organisationen nur Auskunft erteilt, wenn beim Strassenverkehrsamt nachgewiesen wird, dass die Sperrung sie in der Verfolgung eigener Rechte gegenüber der betreffenden Person behindert (§§ 16 und 22 des zürcherischen Gesetzes über die Information und den Datenschutz sowie Art. 126 der Verkehrszulassungsverordnung).

It says something like:

Request for banning vehicle owner data – we want that all our Zurich number plates are banned from disclosure.

Please note: After the ban, public authorities will only get the data if they need them. Individuals and organizations will only get the data if they can proof that they need them (e.g., to file a lawsuit against the vehicle owner).

For me the existence of such a form was very surprising as this sounded to me like an opt-out for sharing your vehicle owner data with the public. After a few more minutes of research I then found out that it is exactly that.

Even worse: There is a website – available to everyone – that lets you query the address data of vehicle owners for all kinds of vehicles (e.g., tractors, motorcycles, cars, trucks – even boats) by the number plate, not only for Zurich, but also for 4 other cantons of Switzerland (Aargau, Lucerne, Schaffhausen, Zug)!

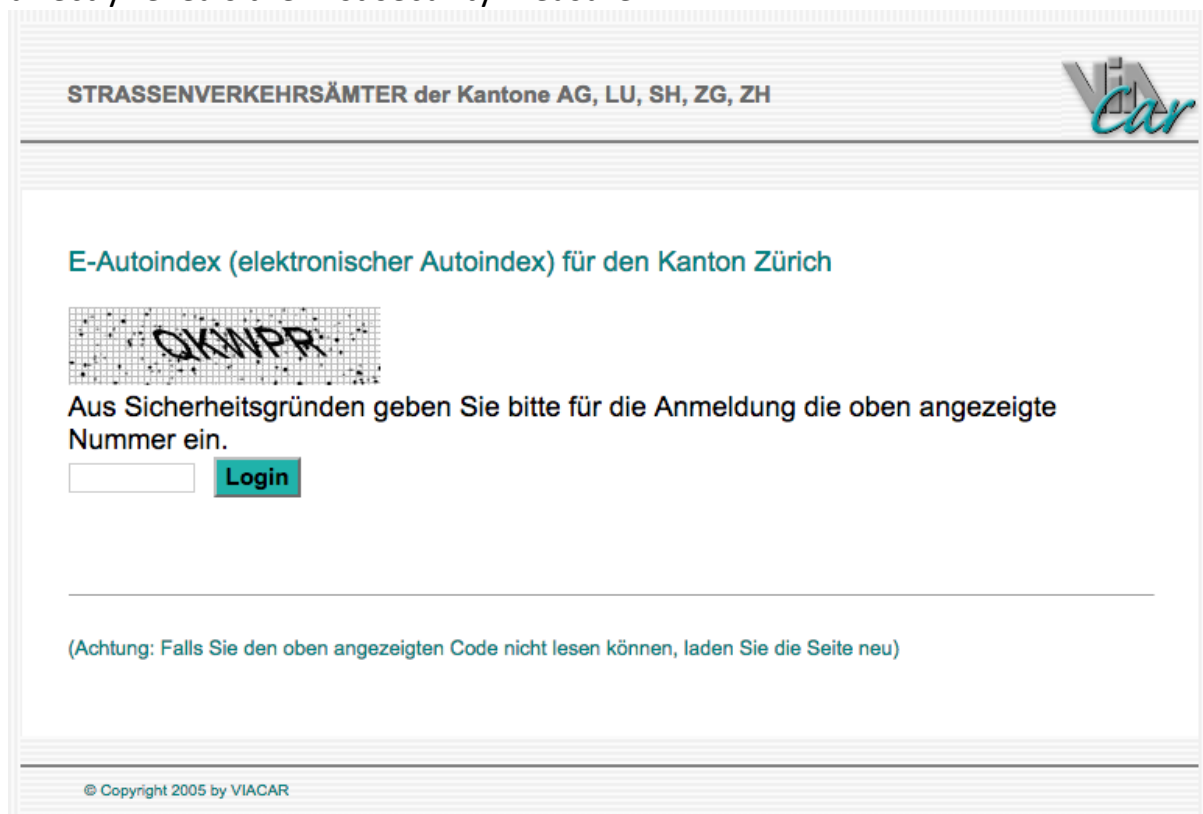
After recovering from the first shock and wondering why such a service must be available to the general public – I really could not come up with just one meaningful reason for that, except that it is a great service for criminals that want to find out, where they have to go to “borrow” the Ferrari they saw on the streets – I decided to take a closer look at the website and check if it at least sufficiently protect its crown jewels, i.e., the vehicle owners address data.

Web Application Overview

The web application is reachable at

<https://www.viacar.ch/eindex/Login.aspx?Kanton=ZH>

where the last two letters define the canton (here: Zurich). The start page directly reveals the first security measure:



The screenshot shows the login page of the VIA CAR application. At the top, the header reads "STRASSENVERKEHRSÄMTER der Kantone AG, LU, SH, ZG, ZH" and features the VIA CAR logo. The main heading is "E-Autoindex (elektronischer Autoindex) für den Kanton Zürich". Below this is a captcha image showing the code "QKINPR". The text below the captcha states: "Aus Sicherheitsgründen geben Sie bitte für die Anmeldung die oben angezeigte Nummer ein." There is a text input field and a "Login" button. A note at the bottom says: "(Achtung: Falls Sie den oben angezeigten Code nicht lesen können, laden Sie die Seite neu)". The footer contains the copyright notice: "© Copyright 2005 by VIACAR".

Before we can query vehicle owner data, we have to solve a captcha, after that we are taken to the number plate input screen:

E-Autoindex (elektronischer Autoindex) für den Kanton Zürich

Suchen Sie den Halter oder die Halterin durch die Eingabe der Kontrollschildnummer im nachstehenden Feld.

ZH

Anzahl Abfragen: 0/5

Here we can enter the number of a number plate we are looking for. In Switzerland number plates have the form

<two letter canton code> <number between 1 – 999999>

As we have chosen the application for Zurich, the two-letter canton code is already given and only a number between 1 – 999999 must be selected. Below the input field there is evidence for a second security measure, it says:

Number of queries: 0/5

So, we can only query the vehicle owner data for 5 number plates. This quota is reset every day and is the same for all cantons except Zug, where it is limited to 3 queries per day.

When we submit the above form, e.g., querying for number plate ZH 1, the following intermediate page is shown:

E-Autoindex (elektronischer Autoindex) für den Kanton Zürich

Ihre Anfrage wird verarbeitet

(Achtung: Falls Sie nach 30s keine Antwort haben, klicken Sie hier.)

Your request is processed...

This page is shown for exactly 30 seconds before the result page loads. So, this is very likely another security measure that aims for limiting the request rate. The result page finally looks like this:

E-Autoindex (elektronischer Autoindex) für den Kanton Zürich

Suchergebnis für ZH 1

[Neue Suche](#)

Art:	Motorrad
Name:	Max Mustermann
Strasse:	Musterstrasse 1
Ort:	8000 Zürich



Art:	Landw. Motorfahrzeug
Name:	John Doe
Strasse:	Doestrasse 2
Ort:	8001 Zürich



Art:	Schiff
Name:	Fritz Blitz
Strasse:	Blitzstrasse 3
Ort:	8002 Zürich

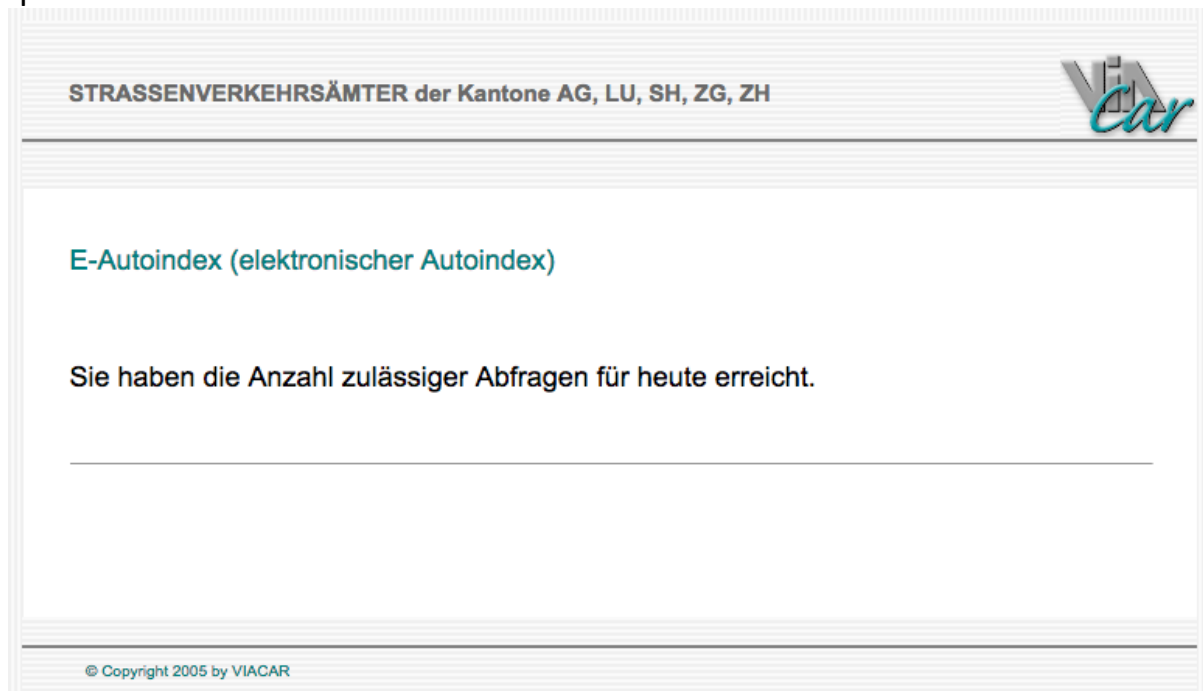
[Neue Suche](#)

Note: The real vehicle owner data was replaced by sample data in order to protect the privacy of the owners.

We can see that there is a motorcycle, a tractor and a boat registered for number plate ZH 1 together with name, street and city of the corresponding vehicle owners. There could be also a registered car for ZH 1, but the owner of

that might be smart enough to hand in the opt-out form mentioned above and thus is not shown in the application.

By clicking the button in the bottom right corner, we could start a new search for another number plate and repeat this until we have reached our daily quota:



The screenshot shows the VIA Car website interface. At the top, the header reads "STRASSENVERKEHRSÄMTER der Kantone AG, LU, SH, ZG, ZH" and features the VIA Car logo. Below the header, the main content area displays the title "E-Autoindex (elektronischer Autoindex)" in blue. A message in black text states: "Sie haben die Anzahl zulässiger Abfragen für heute erreicht." Below this message is a horizontal line. At the bottom of the page, a small copyright notice reads: "© Copyright 2005 by VIACAR".

You have reached the daily limit of requests.

Additionally, there is another security measure that takes effect if 5 minutes have passed since login (i.e., solving the captcha):

E-Autoindex (elektronischer Autoindex)

Die Zeit ist abgelaufen, bitte neu anmelden.

© Copyright 2005 by VIACAR

Time is up, please login again.

When this timeout occurs, we have to go back to the login screen and solve another captcha to continue querying for number plates.

Overall, we have identified four security measures of the web application

1. Captcha
2. Daily quota for queries (3 for Zug, 5 for other cantons)
3. Request rate limitation (30 seconds processing time for queries)
4. Session timeout (5 minutes after solving the captcha)

And now the bad news: It is possible to get around three and a half of them. The following sections describe how.

Bypass the daily quota

When a user visits the website for the first time, he will receive the following cookie:

Set-Cookie: ViaIndZH=Anzahl=0&Date=05.06.2017&de-CH=de-CH;

The cookie *ViaInd*<two letter canton code> contains another two values *Anzahl* (*Count*) and *Date*. The former is initialized with zero and the latter is set to the current date. After the first query for a number plate, with delivery of the result page, this cookie gets updated:

Set-Cookie: ViaIndZH=Anzahl=1&Date=05.06.2017&de-CH=de-CH;

Obviously, the value of *Anzahl* has changed from zero to one and I am pretty sure you might already guess how the cookie looks like after the next query:

Set-Cookie: ViaIndZH=Anzahl=2&Date=05.06.2017&de-CH=de-CH;

It is exactly as it looks like: The daily quota is counted in a cookie on the client! So, no more queries left? No problem, delete the *ViaIndZH* cookie and start again:

Youtube link removed

Note: The video shows daily quota reset using the example of canton Zug (daily query quota: 3, cookie name: ViaIndZG).

Bypass the captcha

The problem with the above approach of resetting the daily quota by deleting the cookies is, that after each reset the captcha must be entered again, which is very annoying, especially if we want to automate the process of grabbing vehicle owner data.

Luckily there is another way to reset the daily quota without having to enter the captcha again: When we edit the *ViaIndZH* cookie instead of deleting it. For that we replace the value of the counter *Anzahl* within the cookie with zero.

So, for example if the cookie is

ViaIndZH=Anzahl=2&Date=05.06.2017&de-CH=de-CH;

We change it to

ViaIndZH=Anzahl=0&Date=05.06.2017&de-CH=de-CH;

Unfortunately, that is not enough to reset the daily quota. It turns out, that this cookie is only read by the web application on the captcha login site. So, editing the cookie does not help us, because after that we have to visit the login site again and solve a captcha, ... right?

Well... this would be the best case in terms of security, but as we already learned in the previous section, the web application puts strong trust in the client and its correct handling of cookies. And it does so not only for quota management, but also when it comes to session and authentication management: After the user has solved the captcha the web application sets another cookie *.AUTOINDEXAUTH* containing a random token, which serves as proof that the user is authenticated. When now the user visits the login page

again you would expect, that this token gets invalidated, so that the user is forced to solve a captcha again in order to obtain a new token. Here is how the web application is trying to achieve token invalidation:

Set-Cookie: .AUTOINDEXAUTH=; expires=Mon, 11-Oct-1999 22:00:00 GMT;

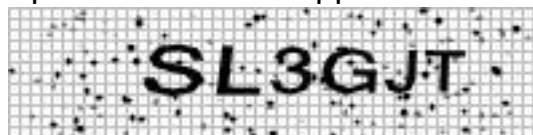
That's it! All the web application does is sending a header that instructs the client to delete the token cookie. The token remains valid on server side! So, all we have to do, after editing the *ViaIndZH* cookie, is opening the login page in a new browser tab and instructing the browser to ignore the above header and voila, the daily quota is reset and we can continue querying in the initial tab without having to enter a captcha again:

Youtube link removed

Bypass the session timeout (i.e., crack the captcha)

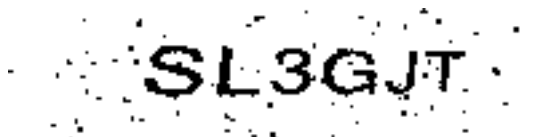
With the two vulnerabilities mentioned above, we could already write a script that automatically grabs vehicle owner data for an arbitrary large amount of number plates, where the attacker has to only solve one captcha manually at the beginning to obtain a valid *.AUTOINDEXAUTH* token. However, as mentioned earlier, the problem is, that these tokens are only valid for 5 minutes after issuance. So, the attacker would have to solve a new captcha every 5 minutes. To make his life easier, it would be therefore great, if we could find a way to automatically solve the captcha and obtain a new valid token after the old one has become invalid.

Here is how a typical captcha of the web application looks like:



Original Captcha

The font color is always black, as well as the pixel noise. There is also always the same grey colored grid in the background. To improve machine readability, we therefore defined a color threshold and set all pixels that are darker to black and all other pixels to white:



Preprocessing step 1: color threshold

To remove the noise, we defined a threshold for the minimum pixel group size and set all pixel groups below that limit to white:

SL3GJT

Preprocessing step 2: pixel group size threshold

Finally, we used the open source OCR engine tesseract (see link section below for details) to solve the preprocessed captcha. With that we were able to achieve success rates of 20%, i.e., we could solve 1 out of 5 captchas automatically. Although this would be enough to achieve the goal of fully automated vehicle owner data grabbing, as there is no limit on how much captchas you can try to solve, we found another flaw that allowed us to increase the success rate even more: When the captcha image is reloaded, the web application generates a new captcha with the same solution!

Youtube link removed

This allows us, to request different image representations of the same solution, solve each of them and apply some statistics in order to find the most probable solution. Even with a very simple statistical algorithm – for each character position take the letter/number that was guessed most frequently by tesseract among all representations – we were able to increase the success rate by 300%.

Mitigate request rate limitation

Unfortunately, we could not find a way to bypass this last remaining security measure of the web application. However, the fact that each query for a number plate takes about 30 seconds processing time, does not limit the number of overall queries one could send per time unit. So, an obvious way of increasing query throughput, and with that mitigate the imposed request rate limitation, is to scale horizontally. For that we wrote the PoC python script *grab-them-by-the-plate.py* that combines all 3 bypass techniques described in the previous sections and supports multithreading. It takes a canton and two numbers a, b as input and grabs vehicle owner data for all number plates of that canton between a and b. So, for example

```
$ python grab-them-by-the-plate.py ZH 1 100
```

grabs the vehicle owner data for Zurich number plates ZH-1, ZH-2, ..., ZH-100. Additionally, one could specify a result file (where vehicle owner data is

written to) using parameter -o and the number of threads to use for querying with -t:

Youtube link removed

With that script, we can really grab them by the plate!

Takeaway

Usually at this point you would find hints how to harden the web application and make it more secure. However, I will refrain from doing that here, because I still have the strong opinion that the world does not need this privacy infringing car theft tool. I recommend filling out the opt-out form *Halterdatensperre* to everybody who has registered a vehicle in one of the affected cantons (see link section for more details). By doing this, at least every individual can make the web application stop sharing his address data. Nevertheless, the big problem remains, that the vast majority of vehicle owners is unaware that their data is publicly available over the internet. I therefore strongly encourage the responsible decision makers in the respective cantons to overthink their decision of making such a web application accessible to the public. If for example someone wants to know the owner of the parked car that is blocking his vehicle, he can call the police or similar state institutions in order to tell him. This procedure works well in other countries and does not require making the owner data of more than one million vehicles worldwide available.

I reported the aforementioned vulnerabilities including a PoC python script confidentially to the web application owner (Viacar AG) in mid-June. Because I have neither received a response from them, nor did they fix any of the issues until end of August, I decided to publish this article, hoping that this will put them under pressure and makes them address the problems. If they continue to stay unresponsive, I will also publish the PoC python script at the beginning of October. All communication details can be found in the responsible disclosure log below.

Links

Halterdatensperre

- Aargau:
https://www.ag.ch/de/dvi/strassenverkehr/fahrzeuge/auskunft_fahrzeugregister/sperrung_halterdaten/sperrung_halterdaten_1.jsp
- Lucerne (form 26 – currently not available):
https://strassenverkehrsamt.lu.ch/Strassenverkehr/online_schalter/formulare
- Schaffhausen:
http://www.sh.ch/fileadmin/Redaktoren/Dokumente/Strassenverkehrsamt/Sperrung_Halterdaten_Antrag.pdf
- Zug:
https://iform.zg.ch/iform/pub/StVA/start.do?generalid=StVA_SPERRE
- Zurich:
https://stva.zh.ch/internet/sicherheitsdirektion/stva/de/StVAfz/FZindex/_jcr_content/contentPar/downloadlist/downloaditems/gesuch_um_sperrung_d.spooler.download.1487751930020.pdf/ZFO010HASP201703.pdf

Tesseract OCR Engine

- <https://github.com/tesseract-ocr/tesseract>

Responsible Disclosure Log

2017-06-11: Detailed vulnerability report including PoC script sent to web application provider (Viacar AG), deadline for first reaction set to 2017-06-30
2017-06-23: Reminder sent to web application provider
2017-06-30: No response received, grace period granted
2017-09-01: Published this article (without PoC script)
2017-10-01: Published PoC python script

Article Version Log

2017-09-01: Initial version