A pile of Mastodung

So, TruthSocial. Valued at ... $8bn. Yes, really. And what do people get for that? Well, allow me to show you. You can safely click on the links in this article, everything pointing to truthsocial[.]com is a snapshot on archive.org.

Open Sores

TruthSocial is running on Open Source. They shared the software they use on their website. The Mastodon file called `mastodon-current.zip` is from **2022-06-08**, and the Soapbox called `soapbox-current.zip` is from **2022-05-13**. Alex Gleason (then Head of Engineering) was keen on keeping things up to date. See this leftover page for another great flashback tot 2022!

Fire and Forget

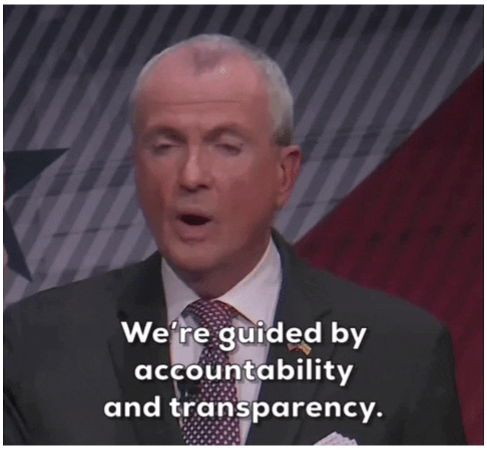
First I had a look at the `soapbox-current.zip` file TruthSocial provided. Soapbox comes with documentation if you want to read along. In the .zip is a file called `instance/soapbox.json` with links to `/mobile` and `/beta`. It used to be the same online, but has since been removed. All still contain the line: '*copyright: "©2022 TRUTH Social"*'. We will see that date turn up everywhere.

A funny aside, both the zipfile and the website contain a file called report.html which is part of soapbox. For an Open Source project it does make sense to have a map of every file of code of Soapbox. But to me it feels somewhat misplaced on a production website supposedly worth $8bn.

I ended up with the feeling TruthSocial's first "Head of Engineering" installed Mastodon, and nobody has done anything but cosmetic changes (and add advertisements!) since. Did I already mention the "Head of Engineering" left? I do wonder If they still have a Head of Engineering. Or even just the digital equivalent of a janitor.

Anyway, let's go dumpster-diving the TruthSocial website.

So transparent you can't see it



First up is the `api/v1/instance` endpoint. I went through the .zip of mastodon they provided. The code that creates that endpoint seems like a cobbled together `app/serializers/rest/instance_serializer.rb`, with certain parts from a Mastodon version newer than or equal to 3.4.2.

The **exact** part which could provide transparency about the number of users (`user_count`) and number of posts (`status_count`) has been disappeared since day one. If I was an investor in ~~this dumpster-fire~~ DJT, that's the two most important numbers I'd want to have. Make of that what you will.

```
def stats
    {
      user_count: instance_presenter.user_count,
      status_count: instance_presenter.status_count,
      [...]
    }
```

The original configuration showed it had a limit of 500 characters per post (excuse me: *truth*), and no support for advertisements yet. The source code shared in the zip also has a maximum of 500 characters and no ads support, so they've made changes since we can't discover looking at the .zip file.

On a throne of rusty cruft

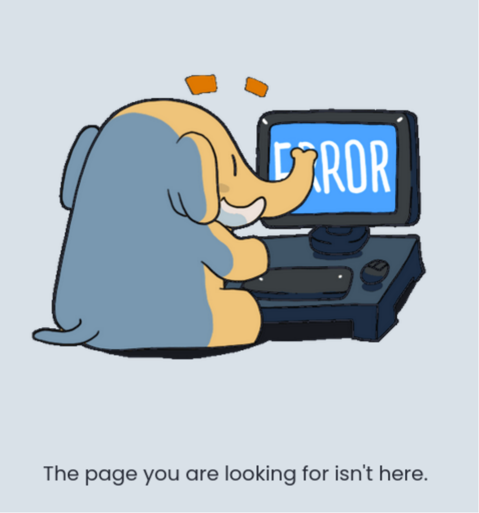
Truth Social's Head of Engineering has a tendency to use old, old software. His current flagship product Soapbox requires a component (Erlang 24) from ... 2021. Luckily it is not End-Of-Life, like some other parts of TruthSocial. But we'll get to that later.

Anyway, Soapbox mentions a "route", `instance/beta` . The route is there, but there is nothing behind it, resulting in an error which exposes the nginx server/proxy used by TruthSocial: `nginx/1.21.6`. It shouldn't come as a surprise this version is also pretty old (2022-01-25), a number of versions behind, and coincidently from around the time TruthSocial was set up.

The `about.example` page mentioned on the Soapbox docs is still there. So is an example configuration (copyright 2020!).

But this is just the supporting software. The core behind TruthSocial is Mastodon. To find out what is running behind the Soapbox facade, I fiddled a little with the TruthSocial website. Nothing even remotely like hacking, just poking around and posting (*truthing*) some messages.

First thing I discovered is that if you call an API url with an invalid path, TruthSocial will show an error page with a cute little Mastodon.



The page you are looking for isn't here.

Inside the html code of that error page is a reference to a stylesheet.

```
<link rel="stylesheet" media="all" crossorigin="anonymous"
href="/packs/css/common-6632dedd.css" integrity="sha256-
aGjPsjGbYVFerGzqX5W/jc1MlRhL7G4+yFgS3DYjvlQ=" />
```

My curiosity got the better of me, so I grabbed the list of all the old instances and collected the `500.html` page of 17316 of them. 50 had the same css file name, but not all were from the 3.4 branch. I have no idea what those four (4) people did to their installs! Anyway, as the table below shows, the `6632dedd` version of common-*blablabla*-.css is most common in the 3.4 branch of mastodon. What is hopefully also apparent is there luckily aren't many 3.4's around anymore.

3.4	**Count**
3.4.0	4
3.4.1	16
3.4.3	1
3.4.4	9
3.4.6	10
3.4.7	2
>3.4	
3.5.19	2
4.2.7	1
4.2.8	1

What we currently know is that this css file is most common in (if not the default of) the 3.4 branch. What we also know is that, unless the people at TruthSocial bothered to manually change the code that shows the version number after they shared the source code, the Mastodon running on their website is indeed version 3.4.1. You can tell from the `"version":"3.4.1 (compatible; TruthSocial 1.0.0)"` part, which is created by the code in `app/serializers/rest/instance_serializer.rb`:

```
def version
    "#{Mastodon::Version} (compatible; TruthSocial 1.0.0)"
  end
```

Go fetch to proof

In normal installations of Mastodon version 3.4, when you create a post with a link, there is code that "fetches" the page that card view of the website. The official code that does this looks like this:

`mastodon-3.4.1/app/services/fetch_link_card_service.rb`

```
Request.new(:get, @url).add_headers('Accept' => 'text/html', 'User-Agent' =>
Mastodon::Version.user_agent + ' Bot').perform do |res|
```

Which returns entries like this:

```
46.4.156.213 - - [25/Mar/2024:00:46:16 +0100] "GET /truth.htm HTTP/1.1" 200 51 "-
" "http.rb/5.1.1 (Mastodon/4.3.0-alpha.3+glitch; +https://infosec.exchange/) Bot"
```

But for truthsocial, the `app/services/fetch_link_card_service.rb` looks like this:

```
Request.new(:get, @url).add_headers('Accept' => 'text/html', 'User-Agent' => "#
{Mastodon::Version.user_agent} Bot").perform do |res|
```

Which creates log entries like this:

```
107.152.38.217 - - [24/Mar/2024:12:02:28 +0100] "GET / HTTP/1.1" 200 1379 "-"
"http.rb/4.4.1 Bot"
107.152.35.98 - - [24/Mar/2024:12:15:27 +0100] "GET / HTTP/1.1" 200 1379 "-"
"http.rb/4.4.1 Bot"
104.192.5.44 - - [24/Mar/2024:12:17:31 +0100] "GET /truth.htm HTTP/1.1" 200 51 "-
" "http.rb/4.4.1 Bot"
107.152.38.167 - - [24/Mar/2024:12:39:42 +0100] "GET /truth2.htm HTTP/1.1" 200 51
"-" "http.rb/4.4.1 Bot"
107.152.35.73 - - [24/Mar/2024:12:47:20 +0100] "GET /thruth.htm HTTP/1.1" 404 125
"-" "http.rb/4.4.1 Bot"
```

Notice that `http.rb/4.4.1`? It is ancient (2020-03-29) and has long since been replaced. The Mastodon 3.4 branch was the last to use http gem version 4.4:

```
mastodon-3.4.0/Gemfile:gem 'http', '~> 4.4'
mastodon-3.4.1/Gemfile:gem 'http', '~> 4.4'
mastodon-3.4.7/Gemfile:gem 'http', '~> 4.4'
mastodon-3.4.8/Gemfile:gem 'http', '~> 4.4'
mastodon-3.4.9/Gemfile:gem 'http', '~> 4.4'
mastodon-3.4.10/Gemfile:gem 'http', '~> 4.4'
mastodon-3.5.0/Gemfile:gem 'http', '~> 5.0'
```

Not sure why they changed this to `#{Mastodon::Version.user_agent}`, as this "interpolation" with the `#` sign is not in any of the other 3.4 versions and clearly breaks the User-Agent string. But maybe there are no mistakes, only happy little accidents? The upside of breaking the Mastodon version string in the User-Agent, consciously or not, is people can't immediately see TruthSocial is running on zombie-software.

Security

Below I'm simply listing the security issues for the 3.4 branch. Figuring out if security issues in more recent versions also existed in the v4.3.1 is left as an exercise for the reader. I have **not** verified or tested whether any of the security issues are patched, because that is shady territory. Please note there are plenty of other parties who are not held back by such considerations. q

According to https://endoflife.date/mastodon, support for the 3.4 branch, specifically for version 3.4.10(!), ended **2022-11-06**. So TruthSocial runs on software that has been out of support or security fixes for–and I'm working from a best-case scenario here– one year and a couple of months. Version 3.4.1 was released 2021-06-03.

v3.4.9

- Fix emoji substitution not applying only to text nodes in backend code (ClearlyClaire)
- Fix emoji substitution not applying only to text nodes in Web UI (ClearlyClaire)
- Fix rate limiting for paths with formats (Gargron)
- Fix out-of-bound reads in blurhash transcoder (delroth)

v3.4.8

- Fix concurrent unfollowing decrementing follower count more than once (Gargron)
- Fix being able to report otherwise inaccessible statuses (Gargron)
- Fix suspended users being able to access APIs that don't require a user (Gargron)
- Fix empty votes arbitrarily increasing voters count in polls (Gargron)
- Fix confirmation redirect to app without `Location` header (Gargron)

v3.4.7

- Fix being able to post URLs longer than 4096 characters (Gargron)
- Fix being able to bypass e-mail restrictions (Gargron)

v3.4.6

- Fix error-prone SQL queries (ClearlyClaire)
- Fix not compacting incoming signed JSON-LD activities (puckipedia, ClearlyClaire) (CVE-2022-24307)
- Fix insufficient sanitization of report comments (ClearlyClaire)
- Fix stop condition of a Common Table Expression (ClearlyClaire)
- Disable legacy XSS filtering (Wonderfall)

v3.4.4

- Fix filtering DMs from non-followed users (ClearlyClaire)
- Fix handling of recursive toots in WebUI (ClearlyClaire)

v3.4.2

- Fix user notes not having a length limit (ClearlyClaire)
- Fix revoking a specific session not working (ClearlyClaire)

So is `ocr/lang-data/eng.traineddata.gz`, available without authentication, If you'd like to cause about 11MB of gratuitous download traffic.

Treasures from the dumpster-fire

Thank you for reading this far, and as a reward, here are two things I found google-dorking.

Tankie Trump

Ads Best Practices & Buyer Roadmap (CONFIDENTIAL, FOR INTERNAL USE ONLY, TMTG, ©2023)