Big Changes are Afoot: Expanding and Enhancing the Have I Been Pwned API

27 OCTOBER 2022

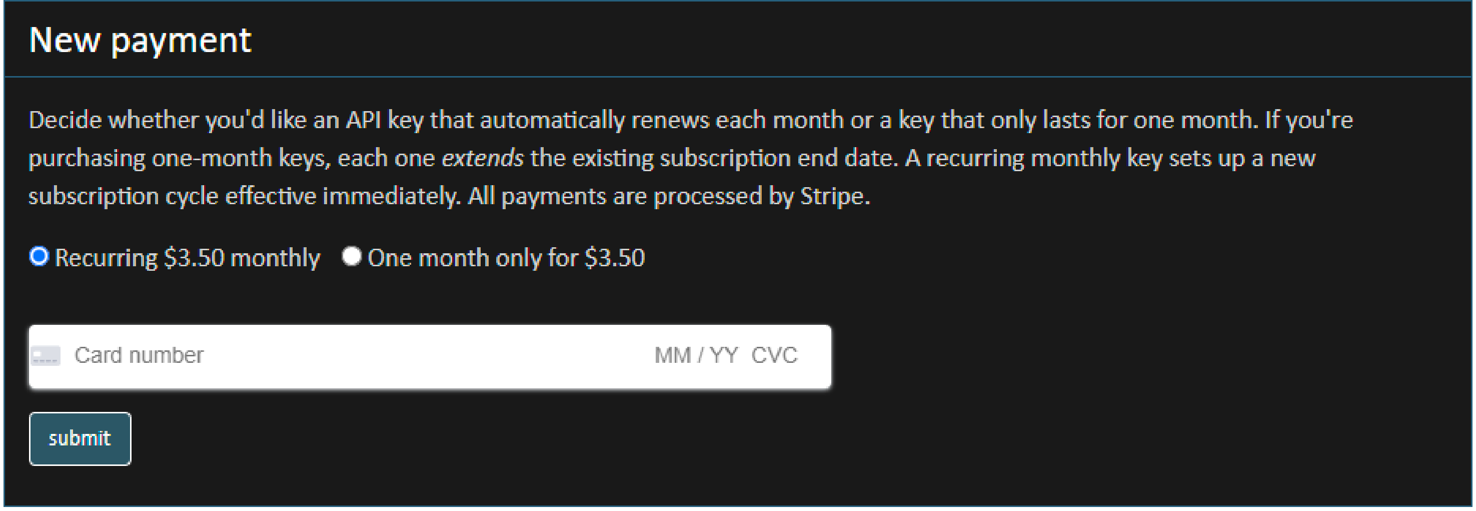
Just over 3 years ago now, I sat down at a makeshift desk (ok, so it was a kitchen table) in an Airbnb in Olso and built the authenticated API for Have I Been Pwned (HIBP). As I explained at the time, the primary goal was to combat abuse of the service and by adding the need to supply a credit card, my theory was that the bad guys would be very reluctant to, well, be bad guys. The theory checked out, and now with the benefit of several years of data, I can confidently say abuse is near non-existent. I just don't see it. Which is awesome 😊

But there were other things I also didn't see, and it's taken a while for me to get around to addressing them. Some of them are fixed *now* (like right now, already in production), and some of them will be fixed very, very soon. I think it's all pretty cool, let me explain:

Payments Can Be Hard… if You Don't Stripe Right

A little more background will help me explain this better: in the opening sentence of this blog post I mentioned building the original authenticated API out on a kitchen table at an Airbnb in Oslo. By that time, everyone knew I was going through an M&A process with HIBP I called Project Svalbard, which ultimately failed. What most people didn't know at the time was the other very stressful goings on in my life which combined, had me on a crazy rollercoaster ride I had little control over. It was in that environment that I created the authenticated API, complete with the Azure API Management (APIM) component and Stripe integration. It was rough, and I wish I'd done it better. Now, I have.

In the beginning, I pushed as much of the payment processing as possible to the HIBP website. This was due to a combination of me wanting to create a slick UX and frankly, not understanding Stripe's own UI paradigms. It looked like this:



New payment

Decide whether you'd like an API key that automatically renews each month or a key that only lasts for one month. If you're purchasing one-month keys, each one *extends* the existing subscription end date. A recurring monthly key sets up a new subscription cycle effective immediately. All payments are processed by Stripe.

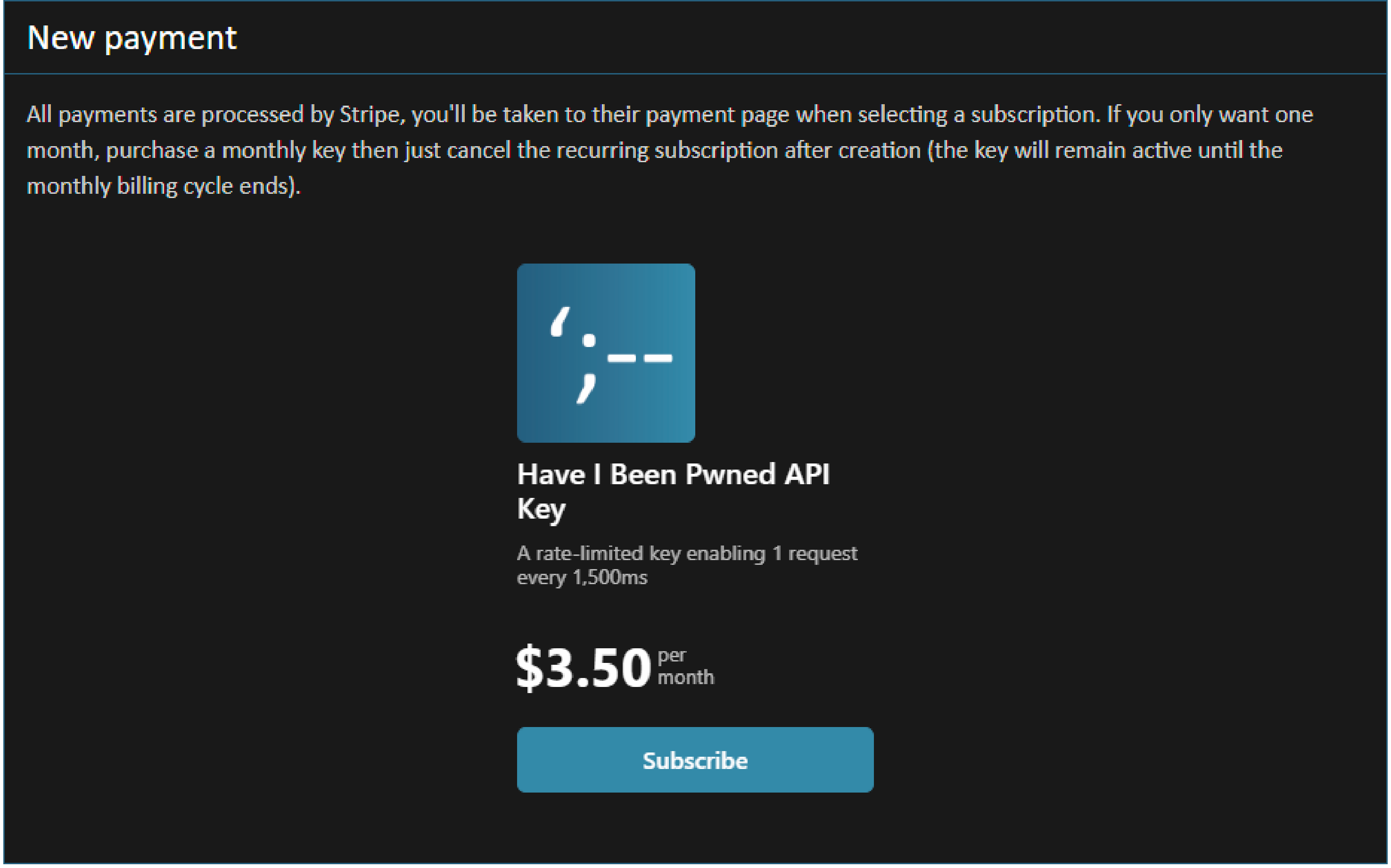
● Recurring $3.50 monthly ○ One month only for $3.50

Cards never ended up hitting HIBP directly, rather the site did a dance with Stripe that involved the card data going to them directly from the client side, a token coming back and then that being used for the processing. It worked, but it had numerous problems ranging from lack of support for things like 3D Secure payments, no support for other payments mechanisms such as Google Pay and Apple Pay and increasingly, large amounts of plumbing required to tie it all together. For example, there were hundreds of lines of code on my end to process payments, change the default card and show a list of previous receipts. The Stripe APIs are extraordinarily clever, but I couldn't escape writing large troves of my own code to make it work the way I originally designed it.

Two new things from Stripe since I originally wrote the code have opened up a whole new way of doing this:

1. Customer Portal: This is a fully hosted environment where payments are made, cards and subscriptions are managed, invoices and receipts are retrieved and basically, a huge amount of the work I'd previously hand-built can be managed by them rather than by me
2. Embeddable Pricing Table: This brings the products and prices defined in Stripe into the UI of third party services (such as HIBP) such that customers can select their product then head off to Stripe and do the purchasing there

Rolling to these services removed a *huge* amount of code from HIBP with the bulk of what's left being email address verification, API key management and handling callbacks from Stripe when a payment is successful. What all this means is that when you first create a subscription, after verifying your email address, you see these two screens:



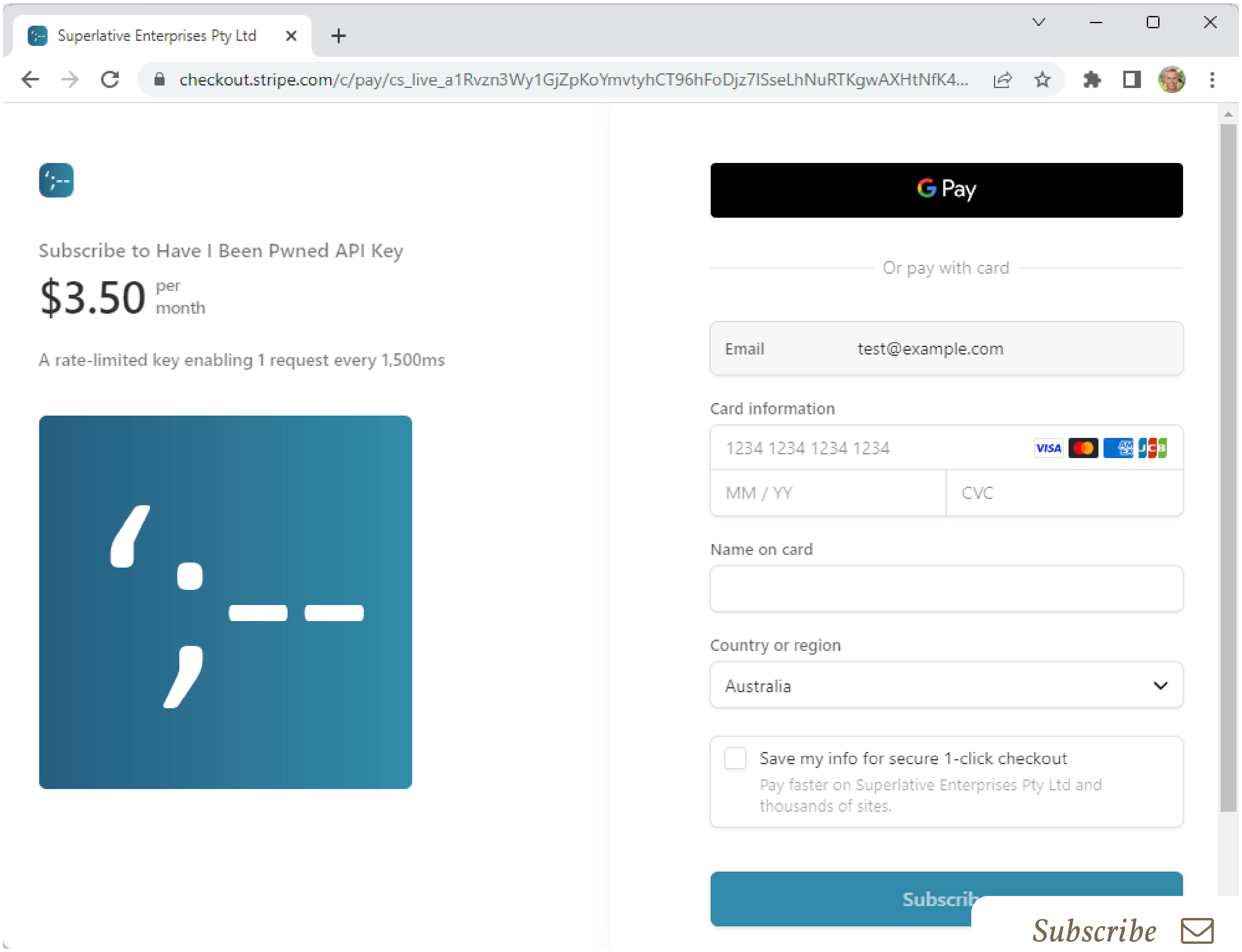
New payment

All payments are processed by Stripe, you'll be taken to their payment page when selecting a subscription. If you only want one month, purchase a monthly key then just cancel the recurring subscription after creation (the key will remain active until the monthly billing cycle ends).

Have I Been Pwned API Key

A rate-limited key enabling 1 request every 1,500ms

$3.50 per month



checkout.stripe.com/c/pay/cs_live_a1Rvzn3Wy1GjZpKoYmvtyhCT96hFoDjz7lSseLhNuRTKgwAXHtNfK4...

Subscribe to Have I Been Pwned API Key

$3.50 per month

A rate-limited key enabling 1 request every 1,500ms

Email test@example.com

Card information

1234 1234 1234 1234

MM / YY

CVC

Name on card

Country or region

Australia

□ Save my info for secure 1-click checkout

Pay faster on Superlative Enterprises Pty Ltd and thousands of sites.

That's the embeddable pricing table following by Stripe's own hosted payment page. I left the browser address bar in the latter to highlight that this is served by Stripe rather than HIBP. I *love* distancing myself from any sort of card processing and what's more, everything to do with actually taking the payment is now Stripe's problem 😊 If you're interested in the mechanics of this, a successful payment calls a webhook on HIBP with the customer's details which updates their account with a month of API key whilst the screen above redirects them over to the HIBP website where they can grab their key. Easy peasy.

I silently rolled this out a week ago, watched it being used, made a few little tweaks and then waited until now to write about it. The rollout coincided with a typical email I've received so many times before:

> “
>
> *First of all I would like to thank you for the wonderful service that helps people to keep track of their email breaches. I was trying to build a product to provide your services via my website, something similar to Firefox, avast and 100's of other companies doing. We were trying to do it according to the guidelines mentioned in the website. However* ***I am not able to renew my purchase due to payment gateway failures at stripe payment.*** *Requesting you to kindly check the same and advise me on alternate methods for making the payment.*
>
> ”

The old model often caused payments to be rejected, especially from subscribers in India. The painful thing for me when trying to help folks is that Stripe would simply report the failed payment as follows:

Subscribe ✉ ×

However, going back to the individual who raised the query above after rolling out this update, things changed very dramatically:

To the title of this section, I simply wasn't "Striping" right. I'm sure there's a way with enough plumbing that it's feasible, but why bother? I cut *hundreds* of lines of code out just by delegating more of the workload back to them. Further, with ever tightening PCI DSS standards (read Scott's piece, interesting stuff) the less I have to do with cards, the better.

This was a "penny drop" moment for me and it's already made a big difference in a positive way. But there's another penny that dropped for me at the same time: one-off keys were an unnecessary problem.

There Are No More One-Off Keys

It was at the moment I was ripping out those hundreds of lines of code that I wondered: why do I have all the additional kludge to support the paradigm of a one-off key that only lasts a month? Why had I built (and was now maintaining) server side code to handle different and UX paradigms to represent one-off versus recurring keys? My gut feel was that most payments formed part of an ongoing subscription but hey, who needs gut feels when you have real data?! So I pulled the numbers:

Only 7% of payments were one-offs, with 93% of payments forming part of ongoing subscriptions.

And so I killed the one-off keys. Kinda, because you can still have a key for only one month, you just purchase a monthly subscription then immediately cancel it via the Stripe Customer Portal:

That's linked into from the API key dashboard on HIBP and it'll take all of 5 seconds to do (also note the ability to change payment method directly on the Stripe site). I've added text to that effect on the HIBP website (you may have spotted that in the earlier screen cap) so in practice, the ability to purchase a one-off key is still there and the main upside of this is that I've just killed a trove of code I no longer have to worry about 🙂 Because this is the internet, I'm sure someone will still be upset, but if you only want a key for a month then that capability still well and truly exists.

All of this so far amounts to doing the same things that were always there but *better*. Now let's talk about the all new stuff!

Annual Billing and Different Rate Limits are Coming… *Very* Soon!

The title is self-explanatory and "very soon" is in about 2 weeks from now 😎

Let me illustrate the first part of that title with a message I received recently:

> *Is there a way to procure a 10 year API key? Our client wants to use the Have I been Pwned plugin for [redacted service name]; however, the $3.50 monthly subscription is too small to go through procurement.*

What's that saying about no good deed going unpunished? In my naivety, I made the pricing low with the thinking that was *a good thing*, yet here we are with that posing barriers! This was a recurring message over and over again with folks simply struggling to get their $3.50 reimbursed. I should have seen this coming after years of living the corporate life myself (I have vivid flashbacks of how hard it was to get small sums reimbursed), and filling out an untold number of expense reports. Speaking of which, this was another recurring theme:

> *Is there a way to pay yearly for HIBP API access vs monthly? Monthly adds overhead in paperwork.*

And again, I get it, this is a painful process. It somehow feels even more painful due to the fact the sum is so low; how much time are people burning trying to justify $3.50 to their boss?! It's painful, and this likely explains why the request for annual payments is the second most requested idea on HIBP's UserVoice. The comments there speak for themselves, and I'm having corporate PTSD flashbacks just reading them again now!

Sticking with the UserVoice theme, the 5th most requested feature is for different pricing on different rate limits. This is mostly self-explanatory but what I wasn't aware of until I went and pulled the stats was just how many people were hacking around the rate limit problem. There are heaps of API accounts like this:

```
hibp+1@domain.com
hibp+2@domain.com
```

```
hibp+3@domain.com
...
```

Because there can only be one key per email address, organisations are creating heaps of unique sub-addressed emails in order to buy multiple keys. This would have been a manual, laborious process; there's no automated way to do this, quite the contrary with anti-automation controls built into the process. Further, each key has it's own rate limit so I imagine they were also building a bunch of plumbing in the back end to then distribute requests across a collection of keys which, yeah, I get it, but man that seems like hard work! When I say "a collection of keys", I'm not just talking about a few of them either; the largest number of active in-use keys by a single organisation is 112. *One hundred and twelve!* The next largest is 110. I never expected that 🤯 (Incidentally, these orgs and the others obtaining multiple keys are all precisely the kinds I want using the API to do good things.)

Building the mechanics of annual billing and different rate limits is only part of the challenge and most of that is already done, the harder part is pricing it. I'm pulling troves of analytics from APIM at present to better understand the usage patterns, and it's quite interesting to see the data as it relates to requests for the API:

There's no persistent logging of the actual queries themselves, but APIM makes it easy to understand both the volume of queries and how many of them are successful versus failed, namely because they exceed the existing rate limit or were made with an invalid (likely that's what I need to work out over the next couple of weeks when I'll launch everything and write it up, as always, in detail 🙂

Summary

The HIBP API has become an increasingly important part of all sorts of different tools and systems that use the data to help protect people impacted by data breaches. The changes I've pushed out over the last week help make the service more accessible and easier to manage, but it's the coming changes I'm most excited about. These are the ones that will make life so much easier on so many people integrating the service and, I sincerely hope, will enable them to do things that make a much more profound impact on all of us who've been pwned before.

Go and check out how the whole API key process works, I'd love to hear your feedback 😊

HAVE I BEEN PWNED

Troy Hunt

Hi, I'm Troy Hunt, I write this blog, create courses for Pluralsight and am a Microsoft Regional Director and MVP who travels the world speaking at events and training technology professionals ⟶

DISCLAIMER

Subscribe ✉ ✖

Opinions expressed here are my own and may not reflect those of others. Unless I'm quoting someone, they're just my own views.

PUBLISHED WITH GHOST

This site runs entirely on Ghost and is made possible thanks to their kind support. Read more about why I chose to use Ghost.

Subscribe