EOTISEC

NATIONAL SECURITY INTELLIGENCE REPORT FOR BUSINESS

EOTISEC Analytical Division

SECTION 1: REPORT IDENTIFICATION

Report Title	OPTIMJACK: Poisoned Optimizely and Third Party Web Scripts Deliver Crypto Theft and iPhone Exploit Routing
Report Number	EOTISEC-2026-085
Date of Report	November 16, 2026
Classification	BUSINESS SENSITIVE // EOTISEC SUBSCRIBER DISTRIBUTION
Originator	EOTISEC Analytical Division
Subject	Attacker JavaScript placed in customer project code served by Optimizely, and in scripts served by Kissmetrics, StorifyMe, and minute-ly, between November 13 and 16, 2026. What is the probability and severity of harm to subscriber websites, their visitors, and their mobile workforces, and what should subscribers do in the next six months?
Customer Sector	Financial Services and Insurance, Retail and Ecommerce, Technology and SaaS, Media and Publishing, Defense Industrial Base

SECTION 2: SCOPE AND PURPOSE

This report answers one question for EOTISEC subscribers: how much risk does the OPTIMJACK campaign pose to organizations that load third party marketing and analytics JavaScript, and to the people who visit those sites, and what should subscribers do about it over the next six months? It covers malicious code observed between November 13 and 16, 2026 inside customer project code served from the Optimizely content delivery network and inside scripts served by Kissmetrics, StorifyMe, and minute-ly, along with the cryptocurrency clipper, payment card skimmer, and iPhone exploit router that code delivered. The source base is open technical reporting by the independent researcher who published as Amirul Cyber and by Hudson Rock, public posts by Scott Helme of Report URI, Optimizely documentation and status pages, and Google Threat Intelligence Group (GTIG) reporting on the Coruna and DarkSword iOS exploit kits. The report does not name an operator, does not estimate dollar losses, and treats the Jscrambler, Confiant, and American Eagle first party bundle side vectors as open questions because no source explains how the attacker reached them. This is initial coverage written while the affected providers were still cleaning their files, so several facts will move in the coming days.

SECTION 3: KEY JUDGMENTS

Each judgment below is a primary analytical conclusion drawn from the source material. ICD 203 probability language applies throughout. Confidence level and explanation follow each judgment. Business Impact callout follows each judgment.

KEY JUDGMENT 1: We assess it is very likely that one operator, or one tightly held criminal toolkit, placed the malicious code into all four providers. Optimizely, Kissmetrics, StorifyMe, and minute-ly assets on unrelated customer sites all pointed visitors to the same command infrastructure and carried the same campaign channel value, and a single BBC project file on the Optimizely CDN served both loader variants on the same day.[1,2]

Confidence: Moderate. *Hudson Rock retrieved file hashes that match archived captures supplied by a separate researcher, and the shared markers are specific to this campaign. The two research efforts share inputs, though, and no affected provider has confirmed or described the intrusion, which holds confidence below High.*

BUSINESS IMPACT: *Any subscriber that loads Optimizely, Kissmetrics, StorifyMe, or minute-ly code on a public website should treat those pages as exposed from November 13 forward until its own logs show otherwise. Pull Content Security Policy (CSP) violation reports, proxy logs, and DNS logs for the indicators in Section 4, compare the Optimizely project code your CDN served against what your team published, and preserve copies for counsel. Retail, banking, insurance, and media sites with heavy marketing traffic carry the most exposure.*

KEY JUDGMENT 2: We assess it is likely that the attacker gained write access at the provider level, through provider infrastructure, publishing credentials, or a connected service that reaches many customer projects, rather than by breaking into each affected customer account one at a time.

Confidence: Low. *No source has established root cause, and Hudson Rock states that the evidence cannot separate provider access from separate customer account compromise. This judgment rests on inference from scale, timing, and the appearance of the same code at four unrelated providers within about four days.*[2,3]

BUSINESS IMPACT: *Do not wait for a vendor root cause before acting. Treat every user, API token, and connected app on your Optimizely account and on any other tag or analytics vendor account as suspect. Review project change history and publication logs for edits you cannot tie to a named employee, rotate API credentials, require single sign on with phishing resistant MFA, and shrink the list of people who can publish project JavaScript. Technology and SaaS firms that run their own customer facing script pipelines should audit those pipelines the same way this week.*

KEY JUDGMENT 3: We assess it is likely that some visitors to affected sites lost cryptocurrency or had payment card data captured while the poisoned code was live.

Confidence: Low. *Delivery of the payload to visitor browsers is well documented. No source has published victim counts, completed card transmissions, or movement of funds into the nine attacker wallet addresses, and Hudson Rock explicitly separates delivery from theft.*[2]

BUSINESS IMPACT: *Banks, card issuers, insurers, and cryptocurrency platforms should expect a tail of fraud tied to cards and wallets used on affected sites between November 13 and 16. Card issuers can add the affected domains to common point of purchase analysis. Exchanges can screen deposits and withdrawals against the attacker addresses listed in Section 4. Merchants that ran any of these scripts on a page that touches checkout should brief their acquirer and their PCI assessor now, because PCI DSS Requirements 6.4.3 and 11.6.1 have required script authorization and tamper detection on payment pages since March 31, 2025.*[4]

KEY JUDGMENT 4: We assess it is very unlikely that iPhones running current iOS 26 releases were compromised through the OPTIMJACK iOS branch.

Confidence: Moderate. *GTIG reports that every DarkSword vulnerability was patched by iOS 26.3 and that Coruna does not work against current iOS.*[5,6] *The attacker router does include an iOS 26 path that no researcher has tested, and that untested branch is the reason confidence is not High.*[2]

BUSINESS IMPACT: *Defense contractors, banks, and any subscriber whose executives or engineers carry iPhones should confirm through mobile device management that every managed device runs the current iOS release and should quarantine devices still on iOS 18 or earlier. GTIG recommends Lockdown Mode where an update is not possible.*[5] *Ask staff whether they browsed the affected sites from an older personal iPhone that also holds work email, chat, or authenticator apps, because that device is the realistic path from this campaign into a corporate network.*

KEY JUDGMENT 5: We assess it is very likely that at least one more compromise of a widely used third party marketing, analytics, or monitoring script will deliver cryptocurrency theft or mobile exploit loaders to commercial websites within the next six months.

Confidence: Moderate. *Three separate script provider compromises in under twelve weeks, Plerdy in late August, DebugBear at Polymarket in late October, and OPTIMJACK in November, show the pattern directly, and the Coruna and DarkSword reporting documents a resale market for iOS exploit chains.*[7,8,5] *The forecast still depends on attacker choices that no source can observe in advance.*

BUSINESS IMPACT: *Every subscriber with a public website should keep an inventory of third party scripts with a named owner for each, enforce a CSP that limits script and connection sources to named hosts with reporting turned on, apply Subresource Integrity (SRI) to every script that does not change by design, and remove experimentation and marketing tags from login, account, and checkout pages. The Polymarket theft, roughly $2.6 million in about fifty minutes through a trusted monitoring script, showed that a host allowlist fails when the trusted file itself changes, so pair it with monitoring of script content hashes.*[8]

SECTION 4: SITUATION AND BACKGROUND

Discovery and Public Disclosure

On November 16, 2026, the researcher publishing as Amirul Cyber released a technical write up naming the campaign OPTIMJACK. The post reported attacker JavaScript inside Optimizely bundles that served a cryptocurrency clipper and payment card skimmer to visitors, documented with fifteen captured page loads from urlscan. Optimizely bundles accounted for twelve of those loads, covering edition.cnn.com, zoom.com, bbc.com, salesforce.com, about.gitlab.com, hootsuite.com, smartsheet.com, virginmediao2.co.uk, stamps.com, tricentis.com, mutualofomaha.com, and insurance.hsbc.com.sg. The remaining three came through other paths: a first party bundle on ae.com, a Jscrambler protected script on wwws.airfrance.it, and a Confiant ad wrapper on foxnews.com. The author wrote that ten more project or snippet IDs across unrelated organizations carried the same injected line.[1]

The same evening, Scott Helme posted that he was tracking a breach of a third party JavaScript host affecting hundreds of high value websites, and later reported 186 sites confirmed infected. In a follow up post he wrote that the affected provider appeared to be aware and working to resolve the problem, that his team could see files being cleaned and their Last-Modified headers changing, and that only 41 of 188 infected script files remained infected at that point.[3] Crypto trade outlets repeated the 186 figure and attributed it to Helme.[9]

Hudson Rock published its own analysis on Infostealers.com the same day and updated it with findings from a researcher identified as Kirk of Ohey. Hudson Rock directly recovered eight poisoned Optimizely project files and a malicious minute-ly loader. Kirk contributed archived response bodies that extended the campaign to Kissmetrics and StorifyMe. The article names 21 affected web properties and states that, for Optimizely, the injection sits inside customer projectJS configurations alongside legitimate experiment code. It states that the initial access mechanism at the providers is unknown.[2]

Observed Timeline (UTC)

The earliest observation in the verified set is a Kissmetrics response on securitybank.com captured at 05:42 on November 13. StorifyMe responses on faz.net, smh.com.au, unibas.ch, and vidaxl.co.uk follow between November 13 and 15. On November 16, a BBC Optimizely response captured at 04:31 loaded a loader from webshell[.]lol, and a later BBC specimen carried the track.js loader instead. Zoom (11:41), GitLab (16:17), Salesforce (14:32), CNN (16:25), Fox Sports through minute-ly (16:26), Mutual of Omaha (18:04), and Fox News through an Optimizely file (18:25) appear in the archive record that day. Hudson Rock retrieved the main payload and further poisoned Optimizely assets between 21:00 and 21:14, and Kirk reported a BBC response without the injection at about 21:10.[2]

How Optimizely Delivers Customer Code

Optimizely Web Experimentation customers place a single script tag in the head of their pages, loaded from cdn.optimizely.com, and Optimizely documentation states that after a customer adds it, the snippet updates automatically. The project ID is the number immediately before the .js in the snippet URL. Optimizely instructs customers not to modify the snippet. The documentation page for the snippet does not mention Subresource Integrity or Content Security Policy.[10] When the snippet loads, it applies queued API calls and runs Project JavaScript, and shared custom JavaScript runs for all variations, including the original.[11] Every customer project file named in the OPTIMJACK reporting follows this URL pattern.[1,2]

Payload Behavior

The injected loader inside affected Optimizely bundles creates a script element that pulls track.js from 1688360[.]cc and appends it to the page, wrapped in error handling. Amirul Cyber described four modules in track.js. The clipper validates and replaces wallet addresses for ETH, TON, BTC, BCH, LTC, TRX, DOGE, and SOL across page text swept every 50 milliseconds, form inputs, links, copy events, clipboard calls, the share API, and fetch and XHR traffic. A collection module sends the page URL, up to 8,000 characters of cookies, a JPEG screenshot built with html2canvas, and matched addresses to /hit.php. The card skimmer validates card numbers, expiry, and CVV, hooks form submits and fetch or XHR POST bodies, and sends results with sendBeacon to /card.php. A fourth module runs only on iPhone, iPad, and iPod.[1] Hudson Rock confirmed nine hardcoded wallet families, found no seed phrase extraction or signing calls, and noted that HttpOnly cookies are out of the payload's reach. BBC and HSBC UAE files also embedded a 24,392 byte Base64 copy of the payload that runs if the remote file fails or takes longer than 2.5 seconds.[2] Two affected sites blocked the loader with CSP, producing a script-src violation, so the payload never ran there.[1]

The iOS Branch

On Apple mobile devices, track.js injects boot.js from two .icu hosts and, on iOS 18.6 or earlier, also adds a hidden iframe.[1] Hudson Rock and Kirk describe a router that sends iOS 13 through 17 to an /A8/ path associated with Coruna, iOS 18 through 25 to a /D11/ path labeled as DarkSword, and iOS 26 to a /C10/ path that was not tested. Kirk tied the kit to a shared dispatcher marker that matches his existing Coruna detection rule and related it to a Caishen iOS C2 V2 ecosystem from earlier Ohey research. Hudson Rock states that the evidence does not establish a zero day, successful device exploitation, or data taken by later stages, and notes that urlscan uses headless Chrome, so its captures show what was served, not what a physical iPhone would do.[2]

Coruna and DarkSword Background

GTIG disclosed Coruna on March 3, 2026 as an exploit kit with five full iOS exploit chains and 23 exploits covering iOS 13.0 through 17.2.1. GTIG first captured parts of it in use by a customer of a surveillance vendor, then saw it in watering hole attacks on Ukrainian sites by the suspected Russian espionage cluster UNC6353, and later in broad campaigns on fake Chinese finance and cryptocurrency sites by UNC6691, a financially motivated actor operating from China. On those sites a hidden iframe delivered the kit to any iOS visitor regardless of location. GTIG described the final payload as stealing financial information, including modules that hook about 19 wallet apps, and wrote that the spread of the kit suggests an active market for second hand zero day exploits. The kit exits if the device is in Lockdown Mode.[6]

GTIG, Lookout, and iVerify disclosed DarkSword on March 18, 2026. It chains six vulnerabilities, three of them exploited as zero days, against iOS 18.4 through 18.7, and GTIG observed it in use by UNC6748 against Saudi users, by customers of the Turkish surveillance vendor PARS Defense in Turkey and Malaysia, and by UNC6353 in Ukraine, with activity dating to November 2025. GTIG reported that all DarkSword vulnerabilities were patched with iOS 26.3, most of them earlier, and recommended Lockdown Mode where updating is not possible.[5]

Related 2026 Incidents

Report URI documented a compromise of the Plerdy analytics script, loaded mostly by Ukrainian sites, that delivered a fingerprinting loader tied to DarkSword between August 27 and September 3, 2026. Report URI cited Sekoia attributing that activity to APT29 with moderate to high confidence and found 33 sites loading the script in its own data during that week.[7] On October 29, 2026, a replaced DebugBear performance monitoring script on polymarket.com drained roughly 28.6 million POL, about $2.6 million, in about fifty minutes. Polymarket's CSP allowed the DebugBear host, so the browser trusted the altered file.[8] Separately, Optimizely disclosed in February 2026 that a voice phishing attack gave an intruder access to internal business systems, including CRM records and back office documents. The company said the intruder could not escalate privileges, install software, or create backdoors.[12] No source connects that incident to OPTIMJACK.

Provider Response as of Report Date

As of the evening of November 16, the Optimizely status page listed no security incident and showed Snippet Delivery, Snippet Building, and the Edge Decider CDN worker as operational.[13] The most recent entry in the Optimizely Security Announcements Center was dated November 2, 2026 and concerned a .NET vulnerability.[14] The cleanup Helme described is the only public sign of provider remediation.[3] EOTISEC found no public statement from Kissmetrics, StorifyMe, minute-ly, Jscrambler, or Confiant.

Indicators of Compromise

The indicators below come from the Amirul Cyber and Hudson Rock reporting. Domains are defanged. Hudson Rock lists the IP address as historical hosting only, not as operator attribution.[1,2]

Indicator	Type	Role
1688360[.]cc	Domain	Payload host and collection server (/track.js, /hit.php, /card.php, /html2canvas.min.js)
webshell[.]lol	Domain	Router loader (/get.js) on Kissmetrics, StorifyMe, minute-ly, and an earlier BBC response
5vobauua3ojpklx[.]icu	Domain	iOS stage (/boot.js, /D11/b.html)
eil786npto9u5bq[.]icu	Domain	iOS stage (/boot.js, /A8/a.html, /JIUJIU888/)
je4ft3gn8vxudum[.]icu, ios-hack[.]icu, l7bcsnwc7nzr3uo[.]icu	Domain	iOS command and telemetry hosts, no completed exchange shown
136.113.106[.]136 (AS396982)	IP	Historical hosting

Indicator	Type	Role
f7b84b9a94f749f8ced3f0112d0a66894e8bbc3b4aa5592021d1b8976a9bad40	SHA256	Live track.js, 29,882 bytes
c2a34c579e4c2e8802752c93b3a9cca0230610f3e4dd6688f4272459c7c7c162	SHA256	Embedded Base64 fallback payload, 24,392 bytes
123f813b18cf5a3b513beabd52a110e19f1c7834f2aa9b972633b2214c57e737	SHA256	webshell[.]lol/get.js
0f2f2a27d19112e5800541daecc91eb6c77e49f9b43ed23c890b876631978533	SHA256	minute-ly loader, 561 bytes
c154f3b644cd51f2446483c7ffd2ae4f178c4b0cb54846cfb5b83215367349cf	SHA256	Kissmetrics i.js loader, 213 bytes
Channel 02111110, window.*k7, DAILI*CFG, data-daili-gate="v1", globalThis.obChTK	Marker	Code markers shared across delivery paths
0xC4026b06D82b0F0B363eda0642D61CfbDc17884e	Wallet	ETH replacement address
19g9rZcPeiz8rUvyWpTRyXXhPnjsWrc3kW	Wallet	BTC replacement address
THvs6p56XkUiMkJeSc51cAP64u2RsTYqxa	Wallet	TRX replacement address
FVtL8NMYS9PA1v3HE67XUGCtGA9JaZUvqav2fU2LrV5n	Wallet	SOL replacement address
UQAgM1VqSZQiJLSG5w-ovurq5CZdxLEK3_7CMmwALTA5e09p	Wallet	TON replacement address
ltc1q3wp7g6ds0ggg6fl6kncpa4sepn0q3qfacgred5	Wallet	LTC replacement address
qzkaaf8n6grmpr0d9lx8xutn3r9lk722f566hhpe58	Wallet	BCH replacement address
DBEGrxRF7KMSuhTS4c1MFviStU6efdMnu5	Wallet	DOGE replacement address

Reporting Discrepancies

The counts differ because the sources measure different things. Amirul Cyber documents fifteen captured page loads, Hudson Rock names 21 properties with recovered or archived evidence and warns that its figures should not be summed, and Helme reports 186 infected sites and 188 infected script files from his own telemetry.[1,2,3] The Fox News vector also differs by source. Amirul Cyber shows the payload arriving through a Confiant ad wrapper, while the Kirk archive records a poisoned Optimizely file on foxnews.com later the same day, so both paths may have been live at different times.[1,2] Hudson Rock states that some later copies of assets lacked the injection and that this does not invalidate earlier archived hashes.[2]

SECTION 5: ANALYSIS

Assumptions

Assumption 1: The poisoned files came from the providers' production delivery hosts and were served to ordinary visitors, not only to scanners. Public urlscan captures on cdn.optimizely.com, matching hashes from separate retrievals, and Helme's count of infected files support this. It fails if a provider shows the captures came from staging paths, lookalike hosts, or scanner only cloaking, which would shrink the exposed visitor population and weaken Key Judgment 3.

Assumption 2: The shared markers, the shared command hosts, and the hardcoded wallet addresses reflect common control of the operation. They do not reflect reuse of a skimmer kit sold to many buyers. The strongest support is that the replacement wallets are fixed in the payload, so every swap pays the same beneficiary, and that one BBC project file carried both loader variants on November 16. It fails if researchers find the same loader and router code in unrelated campaigns with different wallets and different infrastructure.

Assumption 3: The exposure window began on or after November 13. This rests on the earliest verified archive capture. It fails if provider logs or older archives show modified files before that date, which would enlarge the victim pool for Key Judgment 3 and suggest a slower, quieter intrusion than the evidence now shows.

Assumption 4: The operator holds no working exploit chain for current iOS 26 releases. This rests on GTIG patch reporting, on Kirk's iOS 26.3 test that did not load track.js, and on Ohey reporting the C10 path disabled on the hosts checked.[2,5] It fails if a researcher or Apple security note ties the C10 path to a live chain, which would overturn Key Judgment 4.

LINCHPIN ASSUMPTION: Assumption 2 is the linchpin. Key Judgment 1 depends on it entirely, and Key Judgment 2 borrows from it, because a provider level access path only explains four providers if one actor did the work. If the loader and router turn out to be a commodity kit in wide circulation, the case for a single coordinated intrusion collapses and the more likely story becomes several separate compromises sold or shared through a criminal market. Evidence of the same code with different wallet addresses or different command hosts would break it. Revisit this assumption at each production cycle.

Analysis of Alternatives

For Key Judgment 1, the primary alternative is that several unrelated actors bought or copied the same toolkit and hit the four providers separately. We assess this alternative as unlikely. A kit buyer would normally swap in its own wallets and command servers, yet the reported payloads pay into one fixed set of addresses and call one set of hosts, and a single Optimizely project file moved between the two loader variants within hours.

For Key Judgment 2, the primary alternative is that the attacker took over each affected customer's Optimizely account separately, for example with credentials harvested by infostealer malware. We assess this alternative as unlikely. It would require valid sessions into more than a hundred customer projects across four vendors inside a few days, and Hudson Rock, a firm whose core product is infostealer data, reported no infostealer link to this campaign. Absence of that link is weak evidence, which is why Key Judgment 2 carries Low confidence.[2]

For Key Judgment 3, the primary alternative is that no visitor actually lost money or card data. We assess this alternative as unlikely. Most affected pages are marketing and news pages without card forms, and the clipper only fires on pages with payment keywords, which limits yield per site. Yet the code ran for hours to days on sites with very large audiences, four of the affected properties belong to banks and insurers, and the clipper also rewrites clipboard contents, which reaches users who copy an address on one page and paste it elsewhere.

For Key Judgment 4, the primary alternative is that the operator holds and used a working chain against iOS 26. We assess this alternative as very unlikely. The router lists an iOS 26 path, which shows intent, but the only test against iOS 26.3 failed to load the payload at all, and the paths that did serve content map to kits whose vulnerabilities Apple has already patched.[2,5]

For Key Judgment 5, the primary alternative is that the attention this campaign drew, together with provider hardening, deters further script provider compromises for the next six months. We assess this alternative as unlikely. The Plerdy and Polymarket cases were public before OPTIMJACK began and did not prevent it, and the business model of third party scripts, code that changes on the vendor's schedule and runs with full page privilege, has not changed.[7,8]

One structural point runs through all five judgments. Optimizely tells customers the snippet updates automatically and must not be modified, so a customer cannot pin it with an SRI hash the way it can pin a fixed library.[10] That design choice, common across experimentation and tag vendors, is what lets one write at the vendor reach every visitor on every customer page. CSP host allowlisting does not close it either, because the malicious code arrived from the trusted cdn.optimizely.com host. CSP only stopped the attack where the policy also restricted which hosts the injected loader could call next.[1]

Implications by Customer Sector

Financial Services and Insurance: Security Bank, HSBC UAE, HSBC Insurance Singapore, and Mutual of Omaha all appear in the affected set, and a Mutual of Omaha Medicare supplement promotion page was among the captured loads.[1,2] Banks and insurers should check whether any page carrying these scripts also collects account numbers, card data, or identity data, because the skimmer reads fetch and XHR bodies and the collection module grabs screenshots and readable cookies. Fraud teams should flag cards and wallets used on affected domains during the window, and compliance teams should begin a breach notification analysis now, before vendor confirmation arrives.

Retail and Ecommerce: American Eagle, vidaXL, and Stamps.com appear in the reporting, and the American Eagle path ran through its own first party bundle, which means the attacker reached code the retailer hosts itself.[1,2] Any retailer running experimentation code on product or cart pages should assume an assessor will ask for PCI DSS 6.4.3 script inventories and 11.6.1 tamper detection evidence covering these dates.[4] Move experimentation and analytics tags off checkout, set session cookies to HttpOnly, and confirm your CSP restricts connection targets, not only script sources.

Technology and SaaS: Zoom, Salesforce, GitLab, Smartsheet, Tricentis, Hootsuite, and Talkwalker are on the list as victims, and many firms in this sector also ship scripts that run on their customers' sites.[1,2] The second role matters more. Review who can publish to your customer facing script delivery, require hardware backed MFA for that role, alert on any publish outside a change window, and prepare a customer notice template now, because this incident shows that a script vendor's customers learn about a compromise from outside researchers first.

Media and Publishing: BBC, CNN, Fox News, Fox Sports, Frankfurter Allgemeine, and the Sydney Morning Herald served the code to some of the largest audiences in the affected set.[1,2] Publishers carry many third party tags for advertising, video, and audience measurement, and the Fox News and Fox Sports paths show the attacker reaching ad and video vendors, not only Optimizely. Audit every tag vendor with write access to code on article pages, and treat readable cookies on those pages as exposed.

Defense Industrial Base: Public marketing sites at defense contractors often run the same experimentation and analytics tools, and those sites usually sit outside the systems that hold controlled information. The larger exposure is the workforce. Engineers and executives who browsed affected sites on unpatched personal iPhones that also hold corporate email or authenticator apps are the realistic route from this campaign into a contractor network. Enforce current iOS versions through mobile device management, include third party web script providers in supplier risk inventories under NIST SP 800-161 practices, and ask counsel whether any compromised device touched covered defense information before deciding on DFARS 252.204-7012 reporting.

Changes from Prior Assessment

No prior EOTISEC assessment exists on OPTIMJACK or on the compromise of Optimizely customer project code. This report establishes the baseline. Future updates should track vendor root cause statements, victim and loss figures, and any confirmed exploitation of iOS devices through this campaign.

SECTION 6: INFORMATION GAPS AND COLLECTION REQUIREMENTS

For each gap, the format is: Gap statement. Analytical significance. Collection requirement.

1. Gap: How the attacker gained write access to customer project code at Optimizely and to scripts at Kissmetrics, StorifyMe, and minute-ly.

Analytical significance: Key Judgment 2 is most sensitive. A vendor statement confirming platform access would raise it toward very likely, while confirmation of separate customer account takeovers would reverse it.

Collection requirement: Monitor Optimizely, Kissmetrics, StorifyMe, and minute-ly security pages and customer notices, Report URI and Hudson Rock updates, and any regulator filings by the vendors.

2. Gap: Victim counts, completed card transmissions, and funds received by the attacker wallets.

Analytical significance: Key Judgment 3 is most sensitive. On chain evidence of inbound transfers would raise it toward very likely and would support a loss estimate.

Collection requirement: Track the nine replacement addresses on public block explorers, watch card brand compromise alerts, and watch disclosures by affected site operators and data protection regulators.

3. Gap: The behavior of the obfuscated boot.js and whether the iOS 26 C10 path ever served a working chain.

Analytical significance: Key Judgment 4 is most sensitive. Evidence of a working iOS 26 chain would overturn it.

Collection requirement: Monitor GTIG, iVerify, Lookout, Ohey, and Apple security release notes for analysis of these hosts and for new WebKit or kernel fixes credited to related research.

4. Gap: The full list of affected sites and files, and the true start of the exposure window.

Analytical significance: Key Judgments 1 and 3 depend on scope. A start date earlier than November 13 or a list far longer than 188 files would raise the estimated harm.

Collection requirement: Watch Report URI publications, urlscan searches for the indicators in Section 4, and vendor statements that give modification timestamps.

5. Gap: How the Jscrambler protected script, the Confiant ad wrapper, and the American Eagle first party bundle came to carry the payload.

Analytical significance: Key Judgments 1 and 2 are sensitive. Separate access paths at these vendors would point to a broader operation than four providers.

Collection requirement: Seek statements from Jscrambler, Confiant, Air France, and American Eagle, and watch follow up research by Amirul Cyber.

6. Gap: The identity of the operator and its relation to UNC6691 or the Caishen iOS ecosystem.

Analytical significance: Key Judgment 5 is sensitive. A known financially motivated operator with a history of repeated campaigns would firm up the six month forecast.

Collection requirement: Monitor GTIG and Ohey reporting and any law enforcement action tied to the indicators.

SECTION 7: SOURCE SUMMARY STATEMENT

This report draws only on open sources. The technical core rests on two research publications from November 16, 2026: the Amirul Cyber write up, an independent researcher's blog, and the Hudson Rock analysis on Infostealers.com, a commercial threat intelligence vendor.[1,2] These are not fully independent. Hudson Rock credits Amirul Cyber and builds on archived captures from Kirk of Ohey, so corroboration between them is partial. Hudson Rock did recover its own copies of the payload and project files, and those hashes match the archives, which is the strongest independent check in the source base.

Scott Helme's counts of 186 infected sites and 188 infected files are single source and come from his own telemetry, which EOTISEC cannot inspect. X blocked automated retrieval of the posts, so EOTISEC read their text through search index captures of the thread and confirmed the attribution through crypto trade press, which only repeats his figures and adds no original reporting.[3,9] Helme operates Report URI, a commercial CSP monitoring service, and the Report URI blog posts cited for the Plerdy and Polymarket precedents and for PCI dates carry that commercial interest.[4,7,8]

Optimizely documentation, status page, and security announcements are first party corporate communications.[10,11,13,14] They are cited for how the product works and for what the company had and had not said, not as evidence about the

intrusion. The BleepingComputer report on the February 2026 vishing incident relies on Optimizely statements.[12] GTIG reporting on Coruna and DarkSword is vendor research with detailed technical evidence and was published jointly with Lookout and iVerify in the DarkSword case.[5,6] The PCI Security Standards Council blog was used for the scope of Requirements 6.4.3 and 11.6.1, and the Report URI summary for the mandatory date, since the Council post does not state it.[4,15] MITRE ATT&CK and D3FEND entries were confirmed against the MITRE sites.[16,17]

Single source dependencies are the iOS router mapping and the Coruna and Caishen linkage, which come from Kirk of Ohey through Hudson Rock, the side vectors at Jscrambler, Confiant, and American Eagle, which come only from Amirul Cyber, and the site and file counts from Helme. No affected provider has made a public statement about the intrusion. Hacker News threads were reviewed for timing and are not cited. No classified sources were used.

SECTION 8: ICD 503 CONSIDERATIONS AND ATT&CK / D3FEND MAPPING

ICD 503 considerations apply because OPTIMJACK turned an external service, code that a vendor writes and serves, into executable content inside the authorization boundary of every customer website that loaded it. The site owner authorized a vendor, not a specific file, and the file changed without the owner's review. That gap between what was authorized and what actually ran is the core information system risk in this report.

Risk to Information Systems and Data: The payload ran with full page privilege in visitor browsers. It could read and alter page content, form inputs, fetch and XHR traffic, clipboard contents, and any cookie not marked HttpOnly, and it could photograph the rendered page.[1,2] On customer portals, insurance quote pages, and account pages, that reaches personal data, session tokens, and payment data. On iPhones, the router extended the reach to full device compromise on unpatched systems, which places corporate mail, chat, and authenticator apps on personal devices at risk.[2,5,6]

NIST RMF Alignment: The Prepare step should list each third party script vendor as an external system service with a named owner. The Categorize step should rate any page that carries vendor code at the level of the most sensitive data that page touches, not at the level of a marketing site. The Select and Implement steps should add CSP with reporting, SRI where scripts are static, HttpOnly session cookies, and removal of vendor tags from sensitive pages. The Assess step should test that CSP actually limits outbound connection targets. The Monitor step should alert on changes to served script content and on new script or connection hosts. System authorization documentation should record which pages load which vendor scripts and who can publish to them.

Supply Chain Risk: Marketing, product, and growth teams usually buy experimentation, analytics, and tag vendors, and those vendors often never reach the security team's supplier risk inventory. OPTIMJACK shows that each one is a software supplier with the ability to push code into production pages without a change ticket. Defense contractors should place these vendors inside supply chain risk management practice under NIST SP 800-161 and should confirm with counsel whether any compromised endpoint touched covered defense information before assessing DFARS 252.204-7012 reporting obligations. Vendor contracts should require prompt notice of unauthorized changes to customer served code, audit log access, and support for integrity controls.

ATT&CK Technique Mapping

REPORTED marks behavior documented in recovered code or captures. ASSESSED marks analytical inference. No ICS techniques apply.[16]

Technique ID	Technique Name	Tactic	Application to This Report	Status
T1195.002	Compromise Software Supply Chain	Initial Access	Attacker code placed in Optimizely customer project files and in Kissmetrics, StorifyMe, and minute-ly scripts that customer sites load.	REPORTED

Technique ID	Technique Name	Tactic	Application to This Report	Status
T1078	Valid Accounts	Initial Access	Probable use of publishing credentials, API tokens, or a connected service to write project code. Root cause is unconfirmed.	ASSESSED
T1583.001	Acquire Infrastructure: Domains	Resource Development	1688360[.]cc, webshell[.]lol, and randomly named .icu hosts for payload, collection, and iOS staging.	REPORTED
T1608.004	Drive-by Target	Resource Development	Poisoned vendor files turned legitimate, high traffic sites into delivery points.	REPORTED
T1189	Drive-by Compromise	Initial Access	Visitor browsers loaded track.js on ordinary page views.	REPORTED
T1456	Drive-By Compromise (Mobile)	Initial Access	iOS visitors routed to Coruna associated and DarkSword labeled delivery pages. Successful device exploitation is not established.	REPORTED
T1480	Execution Guardrails	Defense Evasion	iOS only branch, iOS version routing, and payment keyword gating of the clipper.	REPORTED
T1027	Obfuscated Files or Information	Defense Evasion	Base64 embedded fallback payload, XOR encoded host strings, and obfuscated boot.js.	REPORTED
T1115	Clipboard Data	Collection	Hooks on clipboard read and write calls and copy events.	REPORTED
T1565.002	Transmitted Data Manipulation	Impact	Wallet addresses rewritten in page text, inputs, and fetch or XHR bodies so payments go to attacker wallets.	REPORTED
T1056.003	Web Portal Capture	Collection	Card number, expiry, and CVV captured from form submits and POST bodies.	REPORTED
T1113	Screen Capture	Collection	html2canvas page images sent to /hit.php.	REPORTED
T1539	Steal Web Session Cookie	Credential Access	Up to 8,000 characters of readable cookies sent to /hit.php. HttpOnly cookies are out of reach.	REPORTED
T1041	Exfiltration Over C2 Channel	Exfiltration	Card data and collected data sent to the same host that served the payload.	REPORTED

Source material is insufficient to map persistence, privilege escalation, or post exploitation behavior on iOS devices, because no source has documented a completed device compromise in this campaign.

D3FEND Countermeasure Mapping

D3FEND Technique	D3FEND ID	Addresses ATT&CK	Application and Priority
Outbound Traffic Filtering	D3-OTF	T1189, T1041	HIGH. A CSP that limits script and connection sources to named hosts, with reporting on, is the one control documented to have stopped this payload. Two affected sites blocked the loader this way. Allowlist cdn.optimizely.com alone and the loader still runs, so the policy must also restrict where loaded code can call.
File Hashing	D3-FH	T1195.002, T1189	HIGH. Apply SRI to every static script and hash monitor served vendor files against the version your team published. Every OPTIMJACK change was a byte level change to a trusted URL, the same failure seen at Polymarket.
Application Configuration Hardening	D3-ACH	T1056.003, T1539	HIGH. Remove experimentation, analytics, and ad tags from checkout, login, and account pages, and mark session cookies HttpOnly. The payload could not read HttpOnly cookies and could only skim cards on pages with card fields.
Software Update	D3-SU	T1456	HIGH for organizations with iPhone fleets. Every exploit chain linked to this router targets vulnerabilities Apple has already fixed, so current iOS removes the documented risk.

D3FEND Technique	D3FEND ID	Addresses ATT&CK	Application and Priority
DNS Denylisting	D3-DNSDL	T1189, T1041	MODERATE. Block the Section 4 domains at resolvers and secure web gateways now. Priority is moderate because the random .icu names suggest the operator can rotate hosts quickly.
Domain Name Reputation Analysis	D3-DNRA	T1583.001	MODERATE. Alert when pages begin calling newly seen hosts. This would have flagged 1688360[.]cc and webshell[.]lol on first contact.
User Behavior Analysis	D3-UBA	T1078	MODERATE. Review vendor console audit logs for publish events by unexpected users or at unusual hours. Value depends on root cause, which is unknown.
Multi-factor Authentication	D3-MFA	T1078	LOW TO MODERATE. Phishing resistant MFA on vendor consoles protects the customer side of the account. It does nothing if the intrusion happened inside the vendor, which the evidence points toward.

These priorities rest on what this campaign actually did: where CSP stopped it, which cookies it could and could not read, which pages it could skim, and which iOS versions its router targeted. They are not generic best practice rankings.[1,2,17]

SECTION 9: ANALYTIC TRADECRAFT SELF-CERTIFICATION

ICD 203 Requirement	Status	Notes
Objectivity: The analysis is free of advocacy, personal preference, or policy bias.	CONFIRMED	*The report flags the commercial interests of Report URI and Hudson Rock and treats Optimizely material as first party. It names Optimizely because the evidence does, without assigning fault the evidence cannot support.*
Independence of Political Consideration: No judgment was shaped to support a particular outcome or audience preference.	CONFIRMED	*Judgments carry Low confidence where root cause and losses are unproven, even though a firmer verdict would read better for subscribers.*
Timeliness: Source material is current and the report addresses matters actionable by the customer.	CONFIRMED	*All primary campaign sources are dated November 16, 2026, the date of this report. Recommended actions can start today.*
Based on All Available Sources: All source material provided was considered. Gaps are documented in Section 6.	CONFIRMED	*EOTISEC searched for vendor statements, status pages, and trade press and found no provider statement. Six gaps are listed in Section 6.*
Source Credibility Described: Source quality is addressed in Section 7.	CONFIRMED	*Section 7 describes the partial independence of the two research sources, the single source counts from Helme, and how the X posts were read.*
Uncertainty Expressed: Probability and confidence language conforms to ICD 203 standards.	CONFIRMED	*Each Key Judgment uses one ICD 203 probability term and a separate confidence statement with rationale.*
Assumptions Distinguished from Facts: All assumptions are labeled. Linchpin assumptions are identified.	CONFIRMED	*Section 4 holds sourced facts. Section 5 lists four assumptions with failure conditions and names Assumption 2 as the linchpin.*
Alternatives Incorporated: At least one alternative hypothesis is addressed for each major judgment.	CONFIRMED	*Each of the five Key Judgments has a stated alternative with an ICD 203 probability assessment.*
Customer Relevance Addressed: Implications for the relevant stakeholder are stated specifically.	CONFIRMED	*Five sector paragraphs name affected organizations in each sector and give concrete actions.*
Clear and Logical Argumentation: The main analytic message is stated up front in Section 3.	CONFIRMED	*Key Judgments precede the evidence, and each Business Impact paragraph states what to do.*

ICD 203 Requirement	Status	Notes
ATT&CK and D3FEND Mapping Completed: REPORTED and ASSESSED attributions are correctly distinguished. Priority designations are grounded in the specific incident.	CONFIRMED	*Only the access path is marked ASSESSED. Each D3FEND priority cites campaign behavior, such as the CSP blocks and the HttpOnly limit.*

SECTION 10: ENDNOTES

[1] Amirul Cyber. "CNN, Zoom, Fox News Served Visitors a Crypto-Stealing Skimmer: OPTIMJACK (Nov 2026)." amirulcyber.com, November 16, 2026. https://amirulcyber.com/2026/11/16/optimjack-optimizely-skimmer/. Accessed November 16, 2026.

[2] Hudson Rock Research Team. "Poisoned Scripts on BBC, Zoom and Other Major Websites Linked to iPhone Exploit Loaders." Infostealers.com, November 16, 2026, updated with findings from Kirk of Ohey. https://www.infostealers.com/article/poisoned-scripts-on-bbc-zoom-and-other-major-websites-linked-to-iphone-exploit-loaders/. Accessed November 16, 2026.

[3] Scott Helme (@Scott_Helme). Thread beginning "We are tracking a breach of a 3rd-party JS host impacting hundreds of high value websites." X, November 16, 2026. https://x.com/Scott_Helme/status/2122422157959471456. X blocked automated retrieval. Text read from search index captures of the thread on November 16, 2026.

[4] Report URI. "PCI DSS 6.4.3 and 11.6.1: The Client-Side Security Guide." report-uri.com. https://report-uri.com/pci-dss. Accessed November 16, 2026.

[5] Google Threat Intelligence Group. "The Proliferation of DarkSword: iOS Exploit Chain Adopted by Multiple Threat Actors." Google Cloud Blog, March 18, 2026. https://cloud.google.com/blog/topics/threat-intelligence/darksword-ios-exploit-chain. Accessed November 16, 2026.

[6] Google Threat Intelligence Group. "Coruna: The Mysterious Journey of a Powerful iOS Exploit Kit." Google Cloud Blog, March 3, 2026. https://cloud.google.com/blog/topics/threat-intelligence/coruna-powerful-ios-exploit-kit. Accessed November 16, 2026.

[7] Scott Helme. "One Hijacked Analytics Script, and Russian iPhone Exploits on Thousands of Websites." Report URI Blog, November 9, 2026. https://blog.report-uri.com/one-hijacked-analytics-script-and-russian-iphone-exploits-on-thousands-of-websites/. Accessed November 16, 2026.

[8] Scott Helme. "Polymarket's CSP Allowed Every Part of the $2.6 Million Crypto Theft." Report URI Blog, October 30, 2026. https://blog.report-uri.com/polymarkets-csp-allowed-every-part-of-the-2-6-million-crypto-theft/. Accessed November 16, 2026.

[9] ChainCatcher. "Bitcoin News: At least 186 websites infected with malware, OPTIMJACK attacks are still ongoing." ChainCatcher, November 17, 2026 (local time, likely UTC+8). https://www.chaincatcher.com/en/article/2305341. Accessed November 16, 2026. Aggregator, cited only to confirm attribution of the count to Helme.

[10] Optimizely. "Implement the one-line JavaScript snippet." Optimizely Web Experimentation documentation. https://docs.optimizely.com/web-experimentation/docs/implement-the-one-line-javascript-snippet. Accessed November 16, 2026.

[11] Optimizely. "Snippet order of activation." Optimizely Web Experimentation documentation. https://docs.optimizely.com/web-experimentation/docs/snippet-order-of-activation. Accessed November 16, 2026.

[12] Sergiu Gatlan. "Ad tech firm Optimizely confirms data breach after vishing attack." BleepingComputer, February 23, 2026. https://www.bleepingcomputer.com/news/security/ad-tech-firm-optimizely-confirms-data-breach-after-vishing-attack/. Accessed November 16, 2026.

[13] Optimizely. "Optimizely Status." status.optimizely.com. https://status.optimizely.com/. Accessed November 16, 2026.

[14] Optimizely. "Security Announcements Center." Optimizely Support, updated November 2, 2026. https://support.optimizely.com/hc/en-us/articles/28863794000909-Security-Announcements-Center. Accessed November 16, 2026.

[15] Alicia Malone. "New Information Supplement: Payment Page Security and Preventing E-Skimming." PCI Perspectives Blog, PCI Security Standards Council, March 10, 2025. https://blog.pcisecuritystandards.org/new-information-supplement-payment-page-security-and-preventing-e-skimming. Accessed November 16, 2026.

[16] MITRE. ATT&CK Enterprise and Mobile technique entries, including T1608.004, T1565.002, and T1456. https://attack.mitre.org/. Accessed November 16, 2026.

[17] MITRE. D3FEND technique entries D3-FH, D3-OTF, D3-DNSDL, D3-DNRA, D3-SU, D3-MFA, D3-ACH, and D3-UBA. https://d3fend.mitre.org/. Accessed November 16, 2026.