

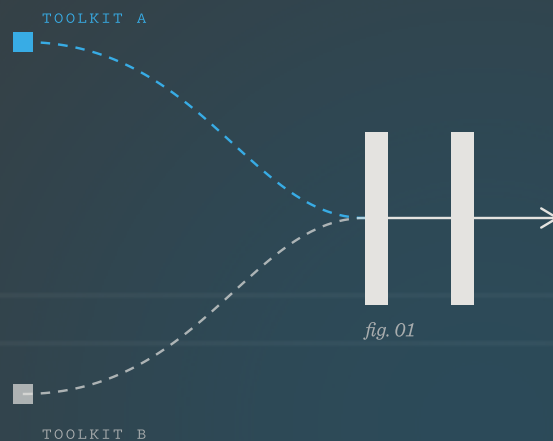
THREAT RESEARCH — CREDENTIAL PHISHING

Two toolkits, one gate

SEPTEMBER 2026

NYASHER FRAMEWORK

WHITEPAPER



Executive *summary*

01

Multi-factor authentication was supposed to be the backstop against credential phishing, but for a campaign we've detected that's been running since mid-August 2026, it isn't holding.

The lure doesn't arrive as a suspicious email; it comes in as a calendar invite. The link isn't a spoof; it routes through Google's own redirector and the CAPTCHA on the landing page is frequently the real Cloudflare Turnstile widget. By the time a target reaches the sign-in screen, the page has already been vouched for by every signal a defender would normally rely on (domain reputation, TLS, Google's own branding).

What's waiting behind that sign-on screen is a live proxy relaying the real Google login flow, second factor included, to an attacker watching in real time. The standard checks built to catch this type of attack assume the adversary has to build their own infrastructure, but that's not how this campaign worked. This whitepaper is an in-depth analysis of the threat.

01 Introduction

A calendar invite arrives from an organization you have corresponded with before, or it arrives via BCC. It carries no attachment and no macro, just a themed calendar invite with a link.

Observed themes include:

- Voicemail
- RFP/RFI with various links to documents
- Court Summons
- Missed Payments

The link resolves through a domain registered the previous week, shows you a Cloudflare verification page and asks you to prove you are human. Sometimes there is even a box to tick. Then you are looking at a Google sign-in screen that is not Google's, on a machine that is being watched.

This campaign steals Google sessions in a way that survives multi-factor authentication, and it has been running continuously since mid-August 2026 while registering as clean in every commercial reputation feed we tested (as of mid-September 2026). The framework behind it is older: public technical writeups from February, June and July 2026 describe the same code, the same backend server, and the same delivery shape, under the name Nyasher, taken from a key hardcoded in the JavaScript. One gate binary, three render modes, N possible origins.

02 Three layers, and only one of them is the operator's

The chain separates into three layers with different owners and different lifespans. Lure domains rotate constantly, harvesting pages persist, and receiving endpoints get replaced mid-campaign. Blocking any single piece changes very little, because each layer is a separate procurement decision. Kit A is an overlay logger on genuine Google markup and Kit B is live MiTM proxy of the sign-in workflow. The kits are deobfuscated in-memory and the malicious code is never a fetchable URL.

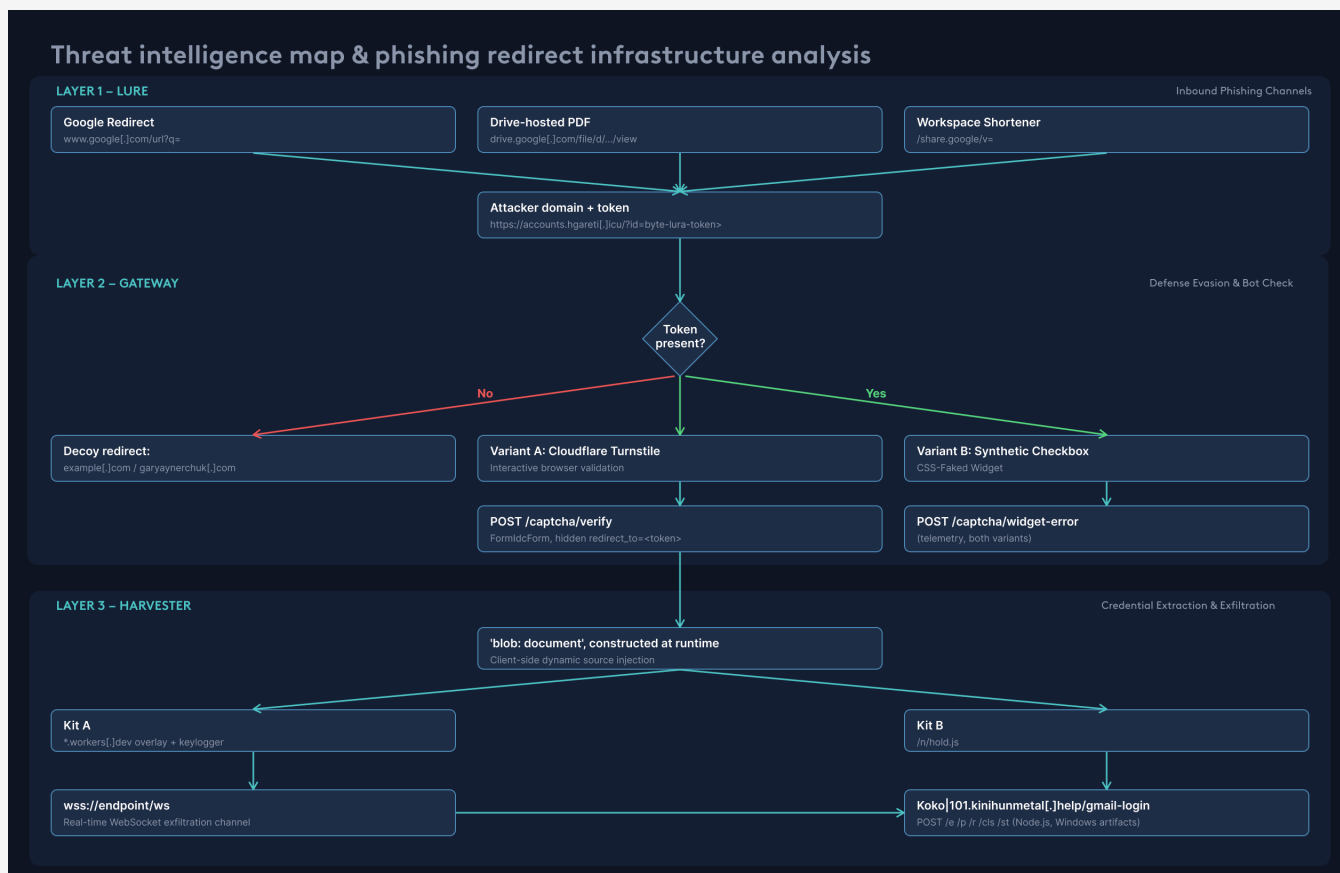


fig 01. – Threat intelligence map & phishing redirect infrastructure analysis

03 Layer One: Google

The link in the invite is not the attacker's URL. It is `www.google[.]com/url`, Google's own redirector, wrapping the lure domain inside `q=`. At the moment the message is scored, the visible domain is Google's, the certificate is Google's, the reputation is Google's, and the domain sits on every allowlist in existence. `share.google`, Workspace's shortener, and direct `drive.google[.]com/file` links do the same job by different routes. Parsing the target of the redirect will sometimes get you to Layer 2, but we've observed intermediary infrastructure abuse, namely `links.notification.intuit[.]com` transaction notification links.

04 Layer Two: The Gate

Each one of these second level URLs carries a 4 byte hexadecimal token that activates layer 2 and tracks the individual user. Navigating to the URL without this token will redirect you to a harmless website. Several of these were observed, with `example[.]com` being an interesting choice, or perhaps a default value.

Two variants run against the same endpoints. One uses the real turnstile challenges.cloudflare[.]com/turnstile/v0/api.js, the other is synthetic, fakes the widget in CSS and computes nothing. Both submit to `POST /captcha/verify`; both report telemetry to `/captcha-widget-error`. Observed form fields on the live gate:

```
<form
id="cfForm"
action="/captcha/verify"
method="POST">
<input
type="hidden"
name="redirect_to"
value="/?deadbeef" <-- URL token example carried from layer 1
style="">
...
```

05 Layer Three: The Harvester

The capture page is constructed as a blob: document injected by the payload. There is no path on the origin to request, so nothing appears in a proxy log as a fetchable URL and no directory listing exposes it; the DOM exists only after the gate. That is why, to our knowledge, the harvest pages have never been scanned: a property of the design rather than an accident of timing. Evidence must be captured at execution time in the browser, and content signatures must be written against the delivered script rather than against a page, since one observed build base64-embeds its entire inner document and injects it at runtime.

06 The keylogger behind the picture

Kit A runs as two Cloudflare Workers on one account, fronted in one case by `audiovoicemail.corneriglen[.]vu/playback/`. It loads genuine Google sign-in markup and overlays invisible input fields on the username and password boxes. The victim sees Google; the keystrokes land in the attacker's fields.

Fetching either Worker returns 2,795,734 bytes of SVG: a 202 by 202 grid of 40,804 one-pixel rectangles, 22,084 of them carrying a four-hex-digit data-c attribute, and no `<script>` element anywhere in the document. Anything inspecting markup sees an image and moves on. Both the inner stages and the operator's own relay interface are recoverable from the container without executing it.

That defense stops at the boundary of anything that runs code. Load the same URL in a browser and the page assembles itself into an in-memory blob: document of roughly 229,000 bytes, six script tags, one of them a 159,467-byte inline script, with every constant present in plaintext: the XOR key, the AES key, the fingerprint field, the operator-console function, and the deployment configuration itself, which decrypts with the published key alone. The container defeats single-pass fetchers and static scanners. It does not defeat Cloudflare's crawler, Google Safe Browsing, or any sandbox with a real browser engine.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Google Voicemail Notification</title>
6      <style>
7        * {
8          margin: 0;
9          padding: 0
10       }
11
12       html,
13       body {
14         width: 100%;
15         height: 100%;
16         display: flex;
17         align-items: center;
18         justify-content: center;
19         background: #000
20       }
21
22       svg {
23         max-height: 100%;
24         max-width: 100%;
25         width: 100%;
26         height: 100%
27       }
28     </style>
29   </head>
30   <body>
31     <svg xmlns="http://www.w3.org/2000/svg" width="202" height="202" viewBox="0 0 202 202">
32       <rect x="0" y="0" width="1" height="1" fill="rgb(2,3,3)" />
33       <rect x="1" y="0" width="1" height="1" fill="rgb(3,3,1)" data-c="385c" />
34       <rect x="2" y="0" width="1" height="1" fill="rgb(0,1,1)" />
35       <rect x="3" y="0" width="1" height="1" fill="rgb(1,1,1)" />
36       <rect x="4" y="0" width="1" height="1" fill="rgb(1,3,0)" />
37       <rect x="5" y="0" width="1" height="1" fill="rgb(3,1,2)" data-c="375c" />
38       <rect x="6" y="0" width="1" height="1" fill="rgb(0,3,1)" data-c="7833" />
39       <rect x="7" y="0" width="1" height="1" fill="rgb(3,2,1)" data-c="5c78" />
40       <rect x="8" y="0" width="1" height="1" fill="rgb(1,2,1)" data-c="7836" />
41       <rect x="9" y="0" width="1" height="1" fill="rgb(1,2,3)" data-c="7832" />

```

fig 02. - The fetched container: 202 × 202 one-pixel rectangles, no script element

The decrypted configuration names where the stolen data goes:

```

playback-xtmw          -> hxxps://gold.sommiedo[.]site      build 10
ref 8882836415
google-playback-audio-aer5 -> hxxps://qwqeq.zyellop[.]online    build 10
ref 8882836415

```

Two Workers, two receiving hosts. `gold.sommiedo[.]site` was created at Namecheap on 2026-06-03 and it is the backend named in the published analyses: four separate campaigns with different lures and different front ends were observed funnelling into `wss://gold.sommiedo[.]site/ws`. A shared backend is the only link the operator cannot rotate cheaply, and it is the only component in the entire chain now carrying an adverse verdict: Cloudflare serves it a phishing interstitial. `qwqeq.zyellop[.]online`, created 2026-09-05, is the opposite. It holds zero certificates, as of time of writing, in transparency logs while serving valid TLS on three subdomains, and answers every path but `/ws` with a LiteSpeed error page.

The overlay carries five anti-analysis measures, all aimed at a person rather than a crawler. A watchdog measures its own timer skew and, if the loop stalls while the challenge is unresolved, empties `documentElement` and redirects to `about:blank`; background-tab throttling and a paused debugger both look identical to that. An `Object.defineProperty` getter is installed on the `id` of a detached element, so expanding it in DevTools navigates the page to `about:blank`. A debugger statement is re-armed every 50 ms through a constructor-built function. The console is nulled across seventeen methods. F12, Ctrl + Shift + I/J/C and Ctrl + U are suppressed, although the console still opens from the browser menu. If you press Ctrl + Alt + Shift + D on the login page, you will get access to a hidden operator console. Some of the fields are set to Removed as the operator disabled external API calls for IP/Region lookups.

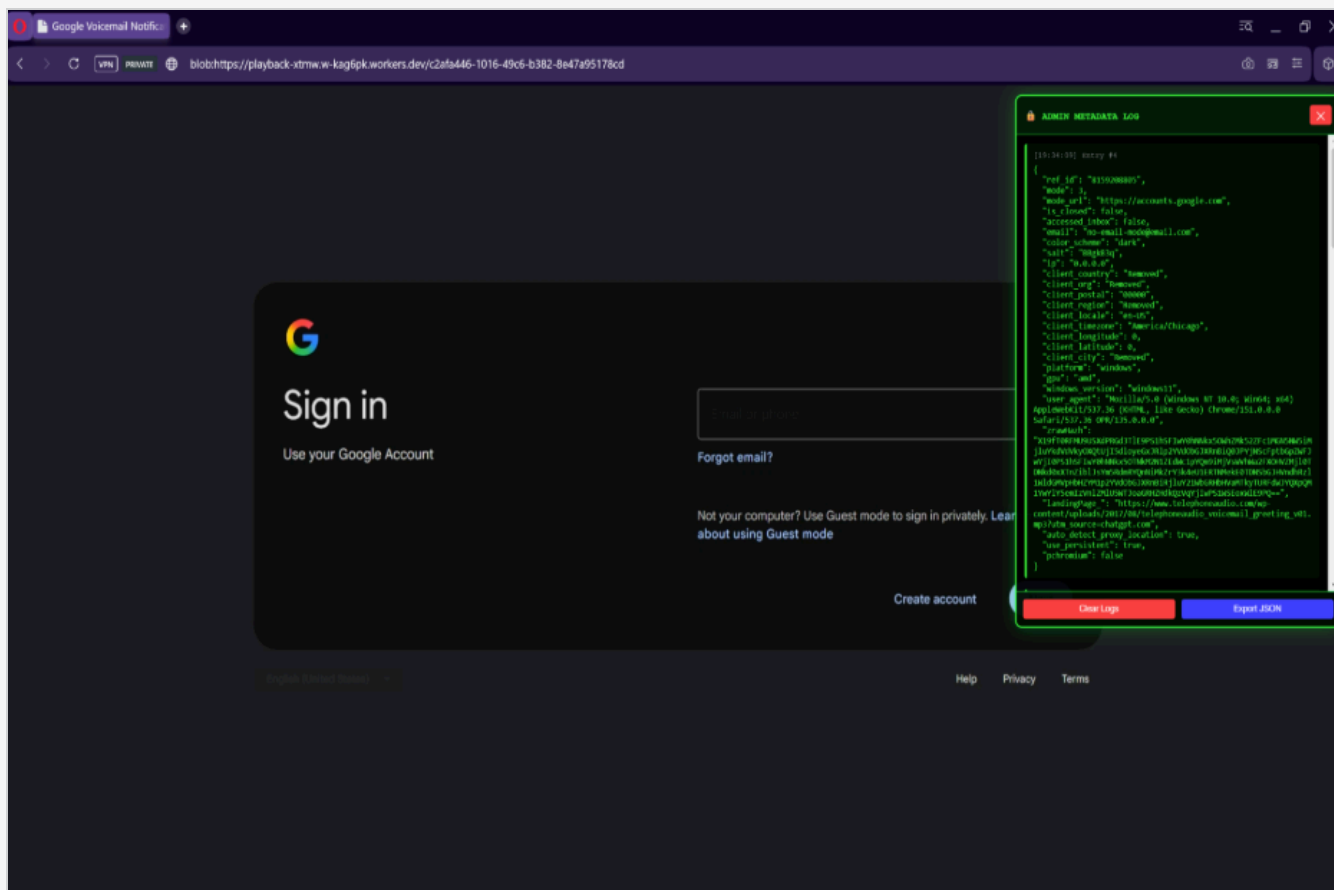


fig 03. - The hidden operator console, opened with Ctrl + Alt + Shift + D

07 The relay, not the socket

The harvester is not fetched directly. The page the gate hands the visitor is an implant that clears the document, paints its own challenge, and then asks a relay Worker what to serve next.

The relay answers a health probe at GET /eal and issues the harvesting document at POST /create. The request carries the victim's browser, the target address, and the lure domain that served the visitor, encrypted with a key derived by hashing a seed compiled into the JavaScript. The list of relays holds one entry in this build but the code randomizes, shuffles and health-checks every candidate before adopting the first that answers. The same build can be driven by any number of lure domains. That relay is a third Worker on the same account, playback-xtmw-2, distinct from the two harvesters. Because the account wildcard covers it, it issued no certificate and produced no transparency log entry, so it could not have been found from CT.

Two details fall out of the relay that are worth more than the harvesters. When the relay is unreachable, when decryption fails, and on every other error path, the visitor is sent to a legitimate hosted form at `form.jotform[.]com/260456798964376`, an inert page that gives the victim no reason to conclude anything went wrong. And when the relay returns a URL rather than inline HTML, the implant retries it five times at 500 ms intervals to let a cold Worker warm up, which produces a distinctive five-request burst from a single visitor.

The two channels do not share crypto. The relay speaks AES-CBC with a SHA-256-derived key and a random 16-byte IV prepended to the ciphertext. The websocket session is AES-256-GCM under a different compiled key.

08 The proxy that eats your second factor

The cloak is not applied uniformly: on four of the lure domains the bare root redirects to `example[.]com` and the full URL with its token is needed to reach anything, but on `hostvipxfl[.]com` the bare root serves the challenge with no token at all. The gate is a synthetic Cloudflare turnstile that accepts any click, and its success screen waits on a hostname, `verify.cloudflare[.]com[.]us`, that has never been registered.

Passing the gate at `hostvipxfl[.]com/in/` loads a 42 KB page that sits inside a live proxy of the real Google sign-in flow. Typing posts to the attacker's server, which forwards to Google and returns the next state. A password submission returned HTTP 524, an origin timeout: the server was waiting on Google. Two other codes on the same host describe the rest of the state. `hold.js` now returns Cloudflare 1020, a zone access rule firing before the origin is reached. The earliest subdomain on the same zone returns 521, origin unreachable. The serving host was replaced mid-observation and the oldest was abandoned in place.

The response vocabulary is the capability list. SMS and authenticator codes, Google's Tap-Yes numeric prompt, phone-based account recovery, and the unenrolled-MFA path are all handled, so a victim completes their own second factor inside a session the operator is observing.

An unhandled exception on the origin discloses `C:\snapshot\dev\src\google.js`, which is the runtime path of a pkg-compiled Node.js binary on Windows. An anonymous POST `/gmail-goback` with the body `{"lid":"0"}` reproduces the trace with no cookie, no Referer and no gate passage. Every response carries `Access-Control-Allow-Origin: *`, so the probe is executable from any page a victim visits.

09 How the infrastructure is bought

Nine lure domains, five registrars, only a single identical nameserver pair twice. Five of them were registered between 1 and 11 September. Individual nameservers recur across the wider set: one pair serves both a July receiving domain and a September one, and another nameserver appears in two different pairs. Cloudflare assigns pairs per zone, so exact matching breaks; overlap on single nameservers does not.

Receiving endpoints follow a construction that can be matched prospectively:

The apex publishes no A and no AAAA record, existing only to hold certificates. Nameservers are delegated to Cloudflare. At least one proxied subdomain publishes an HTTPS record advertising `alpn="h3,h2"` with an ECH parameter decoding to `cloudflare-ech[.]com`. The registration record is amended after provisioning.

Other acquisition habits are harder to disguise. The first certificate for `kinihunmeta[.]help`, a Google Trust Services wildcard, is dated 46 minutes before the domain was registered, so validation authority was held before the name was bought. Registrar-to-CDN intervals run one second, three seconds and 29 seconds on three domains. Every receiving subdomain is provisioned one to twenty days ahead of the wave it serves.

One certificate sits outside the pattern. The `burswel[.]sbs` holds a Sectigo domain-validation certificate, issued 2026-09-13. Everything else in the cluster uses certificates from Cloudflare, Google and Let's Encrypt. A Sectigo account carries its own validation history and billing relationship. It is the least rotatable credential visible here.

The first layer and the third layer are bought differently, and only one of those purchase patterns is stable.

10 What can be said about the operator

Registration data is shielded, and machine indexes hold nothing: every campaign string, every telemetry token and every Turnstile sitekey returns zero results across the largest public scan corpus. The published identifiers are not, however, absent from the human record. The XOR key `_SuperBigNyasher47_` is mentioned in three prior blog posts (but not available in active payloads), and the geolocation API token that this build had commented out with: avoid external API calls had already been published by another researcher, under a different account. At least two such accounts have served this code.

That edit is the clearest behavioral signal in the dataset. Someone removed a third-party dependency specifically because it was observable, in a build compiled after the reporting landed.

WHAT IS MATERIAL DOING ABOUT THIS?

Material actively monitors this campaign and continuously tracks new email variants as the attack evolves. We believe that effective threat detection requires a multi-angled approach: First, detecting the message itself and remediating those messages. This includes targeting the technical indicators of the attack, messaging intent, link destinations and other sources of signal. This allows us to maintain broad coverage across current and future variations. Given the high volume and diverse themes observed throughout this campaign, our strategy has enabled us to detect over 99% (with a reclassification rate of 0.01%) of malicious messages since August 2026 and protect our customers. Second, as we identify the source of the attack, going after the infrastructure of the attacker. For example, Material has worked with Cloudflare to take down some of the infrastructure that the attackers were using. With this multifaceted approach when the attacker pivots, and we believe that they will, Material is well positioned to proactively detect and remediate them and take away some of the infrastructure the attackers depend on.

11 Conclusion

None of this required a zero-day. It required patience and borrowed trust, layered so each piece could fail on its own without exposing the rest: a lure that skips the email filter by not being an email, a link that skips the blocklist because it belongs to Google, a login page that beats MFA by watching you complete it rather than trying to guess it. The one crack in all of it was human: the operator reused a token another researcher had already published under a different account. Everything else about Nyasher was built to look unremarkable at every layer, right up until a browser runs the code. That's the pattern that outlasts this one campaign: infrastructure keeps clearing every check that was built to catch bad infrastructure, so the check that matters now has to happen at execution, not at registration.

Indicators

LAYER	DOMAIN	REGISTRAR	CREATED	ROLE
Lure	biddingselection[.]com	Namecheap	2026-09-11	Turnstile gate; only mail-capable domain in the set
Lure	gcheckonline[.]live	Namecheap	2026-09-11	Turnstile gate
Lure	voicenotemail[.]sbs	Dynadot	2026-09-03	Turnstile gate
Lure	hostvipxfl[.]com	Dynadot	2026-09-01	Kit B gate and Kit B loader
Lure	northernstatemechanical[.]com	1domainregistry	2026-09-08	Unproxied, public redirect service
Lure	world-invitedocument[.]com	Cloudflare	2026-08-19	Nameservers attached one second later
Lure	facilitiesolution[.]com	Hostinger	2026-02-13	NS moved to Cloudflare 2026-08-17
Lure	corneriglen[.]vu	none published	cert 2026-08-07	Entry point into Kit A
Lure	myconsultingconstructionvoice[.]us	none published	cert 2026-09-09	Turnstile gate, deactivated
Kit A host	attached-copy-7tgi.w-swjsii.workers[.]dev	None published	2026-08-13	Harvest Container
Kit B host	kinihunmeta[.]help	NameSilo	2026-08-14	Proxy origin
Receiving	sommiedo[.]site	Namecheap	2026-06-03	Long-lived backend, four campaigns

INDICATORS, CONTINUED

LAYER	DOMAIN	REGISTRAR	CREATED	ROLE
Receiving	longling[.]sbs	NameSilo	2026-07-22	Superseded endpoint
Receiving	zyellop[.]online	Namecheap	2026-09-05	Current endpoint
Receiving	burswel[.]sbs	NameSilo	2026-09-07	Current endpoint, Sectigo outlier
Receiving	bucking[.]top	NameSilo	2026-06-28	Oldest endpoint, recently activated
Decoy	form.jotform[.]com/260456798964376	NA	NA	Decoy URL when worker is down

Timeline

DATE (UTC)	EVENT
2026-02-06	First public writeup of the technique, video-production agencies, Framer proposal template
2026-06-03 18:36	sommiedo[.]site registered at Namecheap; certificates within hours; the long-lived backend
2026-06-28	bucking[.]top registered at NameSilo
2026-06-29	Framework named and documented in public reporting, after its XOR key
2026-07-08	Independent teardown; four campaigns correlated on one backend; geolocation token published
2026-07-22 06:45	longling[.]sbs registered at NameSilo; www certificate the same day
2026-07-24	Cloudflare certificate on longling[.]sbs apex
2026-08-07	corneriglen[.]vu certificate; earliest artifact of this wave
2026-08-10 10:39	longling[.]sbs registration record modified
2026-08-13 15:42	*.w-swjsii.workers[.]dev certificate issued, active Kit A harvester
2026-08-14 19:51	*.kinihunmeta[.]help GTS WE1 wildcard issued
2026-08-14 19:54	ekun.kinihunmeta[.]help certificate issued, 43 minutes before registration
2026-08-14 20:37	kinihunmeta[.]help registered; earliest Kit B artifact, three weeks before Kit A's wave
2026-08-15 06:53	Cloudflare certificate on kinihunmeta[.]help apex
2026-08-17 23:46	facilitiesolution[.]com registrar record changed; nameservers move
2026-08-19 03:11	world-invitedocument[.]com registered; nameservers attached one second later

TIMELINE, CONTINUED

DATE (UTC)	EVENT
2026-08-24 14:34	twist.longling[.]sbs certificate issued
2026-08-27	koko.kinihunmeta[.]help certificate 06:48:32, 101 at 10:55:22; Kit B gains its serving host. Endpoints are staged in small batches roughly four hours apart.
2026-09-01	telephoneaudio[.]com record modified 17:57; hostvipxfl[.]com registered 19:19; first synthetic gate
2026-09-02 09:09	kinihunmeta[.]help record modified, during Kit A's first wave
2026-09-03	voicenotemail[.]sbs registered, nameservers 29 s later; Kit A campaign traffic begins
2026-09-05 22:31	zyellop[.]online registered; no certificate ever published for the zone
2026-09-07	burswel[.]sbs registered 07:58:58, nameservers 07:59:01; second receiving endpoint
2026-09-08 17:15	northernstatemechanical[.]com registered, pointed unproxied at 89.106.200[.]1 (rdns = edge.redirect[.]pizza)
2026-09-11	biddingselecion[.]com and gcheckonline[.]live registered 5 h 11 m apart; the first given a mailbox; first external scan of jihfrs.burswel[.]sbs lands on 404.html
2026-09-13	Sectigo domain-validation certificate on burswel[.]sbs apex
2026-09-14 12:57	Account wildcard *.w-kag6pk.workers[.]dev issued; deactivated Kit A harvester
2026-09-16	Kit B origin trace reproducible anonymously
2026-09-16 19:34	Gate license expires on facilitiesolution[.]com; token path returns a licensing error instead of the challenge
2026-09-17	Kit A endpoints at gold.sommiedo[.]site and qwqeq.zyellop[.]online; reference ID turned over, build counter unchanged. Both Workers render to blob:document exposing the payload. playback-xtmw carries a synthetic challenge with a non-hexadecimal Ray ID. /gmail-ar confirmed fully ungated at 13,215 b with a hardcoded developer address. www.burswel[.]sbs resolves alongside jihfrs. gold.sommiedo[.]site and sage.bucking[.]top confirmed blocked as phishing.