

PolinRider Confirmed Footprint Grows 6.5x Since March

By c0a15726-c5b1-4b0d-85e6-fe15553df9e2

Published: 2026-07-15 · Archived: 2026-07-18 02:01:15 UTC

In the three months since we discovered the PolinRider campaign, we've observed it growing to become a major source of malicious activity. We recently completed a fresh hunt that surfaced an entire new wave of victims: Our July hunt uncovered 2,417 newly-compromised repos, pushing confirmed footprint to 4,367 poisoned repos across 2,152 victims.

The malware still hides in the files developers never read, it still preys on individual job-seekers rather than hardened organizations, and it still spreads through the victims' own GitHub accounts. What this round makes newly clear is the *shape* of the campaign — config files are the front door, the fake-font vector is here to stay, and the victim pool is overwhelmingly personal developer accounts.

Nearly all the repos we discovered in this hunt are net-new: Only 2 repositories overlap with the set we published in April; the other 2,415 had never been seen before. Folded into our earlier findings, that brings PolinRider's cumulative confirmed footprint to 4,367 repositories across 2,152 owners — up from 1,951 / 1,047 in April and roughly 6.5x the 675 / 352 we first documented on March 8. As with our earlier research, we believe this is still an undercount.

Config-file injection is still the workhorse, but the fake-font vector is now firmly established. Of the 2,417 repositories in this round, 2,270 (94%) were hit through config-file injection and 142 through JavaScript smuggled inside a `.woff2` font file — the technique that first appeared as a curiosity in April is now a standing part of the toolkit.

The victim pool is overwhelmingly individual developers. 2,260 of this round's repositories belong to personal user accounts; only 157 belong to organizations. This remains, at its core, a campaign that farms the GitHub accounts of job-seeking developers. If you run take-home coding tests, install unfamiliar frontend templates, or clone project starters from strangers, treat every config and font file as untrusted until you've checked it!

What is PolinRider?

PolinRider is a Lazarus Group (North Korean state-sponsored) campaign that emerged in 2026. Initially it weaponized credentials stolen by [TasksJacker](#), and we consider it a parallel or sub-campaign to [Contagious Interview](#). The threat actor forks popular open source projects and submits malicious pull requests, injecting payloads into JavaScript config files that execute during builds but rarely receive scrutiny in code review. Targets included projects with 1 million+ combined GitHub stars, among them Microsoft VS Code, Apache Superset, Rails, LangFlow, and Expo.

- **Threat Type:** Supply-chain compromise via obfuscated JavaScript injection into developer projects
- **Affected Platform:** GitHub (public repositories)

- **July Round:** 2,417 newly confirmed repositories (2,415 net-new) across 1,219 owners
- **Cumulative Confirmed:** 4,367 repositories across 2,152 owners as of July 9, 2026
- **Primary Vector:** Config-file injection (`postcss.config.mjs` , `tailwind.config.js` , `eslint.config.mjs`), with a growing fake-font (`.woff2`) secondary vector
- **Attribution:** Lazarus Group / DPRK, continuing from Contagious Interview and TasksJacker operations

See the [#polinrider tag](#) for associated threat reports, and check out these blogs to understand the history of the campaign:

Campaign scope: the growth curve

The important thing to understand about this round is that the 2,417 repositories are *newly discovered* — they are not a re-count of what we already knew. Cross-referencing against the April dataset, only 2 repositories appear in both; the other 2,415 are entirely new. So rather than the campaign inching up from 1,951 to 2,417, the correct reading is that April's confirmed set and this round's confirmed set stack on top of each other:

Metric	March 8	April 11 (cumulative)	July 9 (cumulative)
Confirmed repos	675	1,951	4,367
Unique owners	352	1,047	2,152
New repos found in round	675	+1,276	+2,417

From April to July, the cumulative confirmed repository count grew by **2,416 (+124%)** and owners by **1,105 (+106%)**. Since March, confirmed repositories are up roughly **6.5x**. This is not a campaign that is winding down — the July round found more brand-new victims than the entire footprint we knew about in April.

The same caveats from April still apply, and they still push the real number higher than what we can confirm: GitHub API pagination limits, Sourcegraph indexing lag, deleted and privatized repositories, and infections on non-default branches all keep confirmed victims below the true total.

July 2026 attack vector analysis

The July round is the first time we can put clean numbers on *how* victims are being hit, rather than just the number of victims. Each newly-confirmed repository is tagged with its injection vector and the specific file the payload landed in. The vast majority contain a contain injection, while a small

- **93.9% Config-file injection:** Dropping the obfuscated loader into a build config that runs automatically when the project is installed or opened
- **5.9% Fake-font vector:** The same JavaScript payload is smuggled inside `public/fonts/fa-solid-400.woff2` (a file no developer ever opens or reviews), has settled in as the established second technique.
- **0.2% Both:** The handful of repositories carrying *both* a config injection and a TasksJacker `.vscode/tasks.json` payload are the clearest on-disk evidence of the campaign merger we documented in April.

Vector	Repositories	Share
Config-file injection	2,270	93.9%
Fake-font (<code>.woff2</code>)	142	5.9%
Config-file + TasksJacker (both)	5	0.2%

The specific files being weaponized tell the same story. They payload lands in the parts of a JavaScript/TypeScript project that developers copy from a template and never read:

Hit path	Repositories
<code>postcss.config.mjs</code>	1,135
<code>tailwind.config.js</code>	213
<code>eslint.config.mjs</code>	195
<code>public/fonts/fa-solid-400.woff2</code>	141
<code>frontend/postcss.config.mjs</code>	79

Hit path	Repositories
<code>next.config.mjs</code>	42
<code>frontend/tailwind.config.js</code>	38

PostCSS, Tailwind, and ESLint configs together account for the large majority of hits. These are files that ship with nearly every modern Next.js / React frontend, run as part of the build, and are almost never inspected line-by-line during code review...which is exactly why the actor keeps returning to them.

Most victims are individual developers

The owner breakdown for this round reinforces that this is a campaign aimed at people, not organizations:

- **90.4%** (1,102) are individual user accounts vs. **9.6%** (117) organizations
- **93.5%** (2,260) are user-owned repositories vs. **6.5%** (157) organization-owned

The most-affected accounts each carry a large cluster of compromised repositories. This pattern is consistent with a developer who ran one malicious template, got infected, and then unknowingly pushed the payload across their entire personal portfolio:

Owner	Affected repos
<code>Dominion116</code>	25
<code>XaidenLabs</code>	17
<code>rayhanalmim</code>	16
<code>alamin-sujon</code>	15
<code>nimurr</code>	14

Owner	Affected repos
Farhanasharna2000	14

On the organization side, `FSDTeam-SAA` (10 repos), `TerraDharitri` (6), and `hngprojects` (5) are the largest clusters. `FSDTeam-SAA` is the same account we flagged in April as carrying *both* PolinRider config injections and TasksJacker `.vscode/tasks.json` payloads. This remains one of the cleanest examples of the two clusters operating as one.

The first-seen dates on these repositories span from as far back as 2017 up to July 1, 2026. The old dates aren't evidence of a decade-long campaign — they're legitimate projects that were created years ago and only recently compromised, or forked and back-injected. The volume is concentrated across 2025 and the first half of 2026, which is consistent with everything else we've observed.

Technical indicators (unchanged from April)

The payload internals have not changed since our April writeup. Both the original and rotated variants are still in circulation, and the detection markers remain valid:

Original PolinRider variant:

- Signature marker: `rmcej%otb%`
- Decoder function: `_$_1e42`
- Injected globals: `global['!']` , `global['r']` , `global['m']`

Rotated variant:

- Signature marker: `Cot%3t=shtP`
- Decoder function: `MDy`
- Injected globals: `global['_V']` , `global['r']` , `global['m']`

Both variants share the same blockchain dead-drop infrastructure (TRON, Aptos, BSC) with unchanged XOR keys, and both use the same file-injection architecture described in the previous post. The five Vercel C2 subdomains and the StakingGame template UUID we published in April are still the relevant network and template IOCs.

Detection

If you develop in JavaScript/TypeScript, run these against your projects. They are the same checks from April, and are still the fastest way to know if you've been hit:

```
# Original variant
grep -rE "rmcej%otb%|_$_1e42|global\[!''\]" ~/projects

# Rotated variant
grep -rE "Cot%3t=shtP|function MDy\(f\)|global\[ '_V'\]" ~/projects

# Fake-font vector – inspect any font that is suspiciously large or recently changed
find ~/projects -name 'fa-solid-400.woff2' -exec grep -lE "rmcej%otb%|Cot%3t=shtP" {} \;

# Config files are the primary target – audit these across every project
find ~/projects \( -name 'postcss.config.mjs' -o -name 'tailwind.config.js' -o -name 'eslint.config.mjs' \) \
  -exec grep -lE "rmcej%otb%|Cot%3t=shtP|_$_1e42|function MDy" {} \;

# Propagation artifact
find ~/projects -name 'temp_auto_push.bat'

# StakingGame template UUID
grep -r "e9b53a7c-2342-4b15-b02d-bd8b8f6a03f9" ~/projects
```

Indicators of Compromise (IOCs)

Signature markers

```
rmcej%otb%
Cot%3t=shtP
```

Decoder functions

```
_$_1e42
MDy
```

Primary target files

```
postcss.config.mjs
tailwind.config.js
eslint.config.mjs
next.config.mjs
public/fonts/fa-solid-400.woff2
```

Propagation artifact

```
temp_auto_push.bat
```

Template UUID

```
e9b53a7c-2342-4b15-b02d-bd8b8f6a03f9
```

If you find any of the markers above in your projects, assume full compromise of that development environment:

1. Delete the affected directories and rotate **all** secrets: npm tokens, AWS/GCP credentials, GitHub PATs, and SSH keys.
2. Audit persistence: SSH `authorized_keys` , crontabs, launchd agents (macOS), systemd services (Linux), and scheduled tasks (Windows).
3. Check browser password stores and crypto-wallet access times.
4. Review your GitHub account for unauthorized pushes — the campaign spreads by pushing the payload back out through victim accounts.
5. Scan for InvisibleFerret Windows artifacts (detailed in the reverse-engineering section of the previous post).
6. Report anything you find to [OpenSourceMalware.com](https://opensourcemalware.com).

Source: <https://opensourcemalware.com/blog/polinrider-blast-radius-grows>