

ComplyCrawl vs Scrapy / Playwright / Crawlee

Date: 2026-05-25

Buyers comparing ComplyCrawl to existing crawlers ask the same questions. Here are the answers.

The short version: **ComplyCrawl is not faster, not more flexible, and not better at JavaScript-heavy pages than Playwright.** What it is, uniquely, is *auditable*. Every fetch is gated by robots.txt enforcement, logged in a tamper-evident hash chain, and exportable as a PDF a compliance auditor will accept.

Capability	Scrapy	Playwright	Crawlee	ComplyCrawl
Crawl static HTML at scale	✅ Best	⚠️ Slow	✅	✅
Render JavaScript	❌	✅ Best	✅	⚠️ Limited
robots.txt parsing	⚠️ Optional	❌	⚠️ Optional	✅ Enforced by default
Consent gating (refuses to run without explicit config flag)	❌	❌	❌	✅
Tamper-evident audit log (SHA-256 hash chain over every fetch)	❌	❌	❌	✅
Encrypted artifact storage (Fernet)	❌	❌	❌	✅
Branded PDF audit report	❌	❌	❌	✅
GDPR / SOC 2 control crosswalk doc	❌	❌	❌	✅ (see GDPR_MAPPING.md)
Ready playbooks (SEO regression, catalog monitor, link health)	❌ DIY	❌ DIY	❌ DIY	✅ Three included
Active development / large community	✅ Best	✅ Best	✅	⚠️ Solo-maintained

Capability	Scrapy	Playwright	Crawlee	ComplyCrawl
Free / open source	✓	✓	✓	✗ Source for sale

When you don't need ComplyCrawl

- You're crawling sites where compliance is not a buyer or audit concern → Scrapy or Crawlee, save the money.
- You need to scrape JS-heavy single-page apps → Playwright is the right tool.
- You want a free, community-supported library → ComplyCrawl is not it.

When ComplyCrawl pays for itself

- Your customer signed a DPA / DPIA that references active web monitoring. You need to produce an audit log on demand.
- You're a marketing/SEO agency and the client's legal team asks how you ensure robots.txt is respected.
- You operate in a regulated vertical (finance, healthcare, legal) where "we ran a Python script" is not a defense.
- You want to white-label compliance-grade monitoring under your own brand — saves you 4–8 weeks of building the audit/encryption/reporting layer.

What is uniquely defensible about the design

1. **Default-deny.** A misconfigured ComplyCrawl run does *nothing*. There is no `--ignore-robots-txt` flag. This is intentional; safety is enforced by absence of an escape hatch.
2. **Append-only hash-chained log.** Mutating a past fetch entry breaks every subsequent hash. Standard pattern in financial audit logs; rare in crawlers.
3. **Encrypted at rest.** Fetched artifacts are encrypted with Fernet (AES-128) before persistence. The vault key is separate from the run config.
4. **Reproducibility.** Every run records the config snapshot. Given the same config and a static snapshot of robots.txt, the run output is reproducible.

Real sample outputs

The repo ships with `demo_output/` containing real artifacts produced by running each playbook:

- `seo_findings.csv` — SEO regression detector output
- `catalog_findings.csv` — catalog change detection
- `link_findings.csv` — broken link audit
- `report.pdf` — branded PDF compliance report (this is what buyers' clients see)
- `run_summary.csv` — top-line run statistics

These are not mockups. Run `make demo` to regenerate them yourself.

Honest limitations

- JavaScript rendering is via subprocess to a headless Chrome; slower and more memory-hungry than Playwright-native.
- The hash-chain is in-process — for cryptographic non-repudiation at audit time you'd want to anchor periodic chain roots to an external timestamp service. Not included.
- Solo-maintained; if you need a Slack channel and 24/7 support, Scrapy's commercial offerings exist.
- Code only. No SaaS, no hosted version, no managed service.