AWFFull 4.0.0 — Modernising a 2008 Web Log Analyser

2026-03-12

AWFFull 4.0.0 brings a long-overdue overhaul to a C-based web server log analyser forked from Webalizer: PCRE2 replaces EOL libpcre, libmaxminddb replaces the deprecated GeoIP .dat format, new ASN statistics show traffic by network operator, and optional Intel Hyperscan, mimalloc, and XXH3 deliver measurable performance gains on large logs. The release also adds a full GitLab CI/CD pipeline, .deb/.rpm packages, a multi-stage container image, updated browser and bot detection lists, and 100%-complete translations for 6 languages.

AWFFull 4.0.0 — Modernising a 2008 Web Log Analyser

Background

AWFFull 4.0.0 is a web server log analyser. It reads Apache/nginx/Squid access logs and produces HTML reports with graphs showing traffic, top URLs, referrers, user agents, country statistics, and more. It was originally forked from The Webalizer by Bradford L. Barrett (1997–2001), extended as AWFFull by Steve McInerney from 2004, and actively developed until 2008, when it was last released as version 3.10.2.

It still works well in 2026 — the core log-parsing logic is solid and the output is clean. But the codebase had accumulated 18 years of code: EOL libraries, dead links, obsolete command-line flags, a broken GeoIP implementation, and a very old `config.guess`. Time for a proper cleanup.

This post covers what changed in AWFFull 4.0.0 and why.

Is a Static Web Log Analyser Still Useful?

Good question. The short answer is: yes — and arguably more so than before.

Modern web analytics platforms — Google Analytics, Matomo, Plausible, Grafana with Loki, ELK stacks — are excellent for live dashboards and drill-down exploration. But they share a common weakness: **they are slow when you need to process large volumes of historical log data and render the results as a complete set of graphs and tables**.

Feed a year of Apache logs (say, 500 million lines across 50 GB of gzipped files) into an ELK pipeline and you'll be waiting a long time for the data to be ingested, indexed, and ready to query. Then wait again each time you open a dashboard. Grafana rendering a year's worth of nginx traffic across 30 panels? Expect seconds of load time per page, per query.

AWFFull's approach is different: it processes the raw log files directly, keeps all state in a compact binary history file, and writes **static HTML + PNG output** that loads instantly in any browser — even on a machine from 2010, over a slow connection, or offline entirely. There is no query layer, no index, no database, no running server. The report is just files.

For the typical use case — a sysadmin or developer who wants a monthly overview of who accessed their server, from where, with what agents, hitting which URLs — this is exactly the right trade-off:

- **Fast to generate**: processing a month of logs takes seconds, not minutes
- **Instant to view**: static HTML, no round-trips, no query engine
- **Self-contained**: the report directory can be archived, moved, or served from any static file server
- **Privacy-friendly**: no third-party JavaScript, no tracking pixels, no data leaving your infrastructure — the raw IP addresses stay in your own logs
- **Works on air-gapped systems**: nothing requires internet access at runtime

It also complements rather than replaces live monitoring. You would typically use Prometheus/Grafana (or similar) to watch your infrastructure in real time, and AWFFull to produce the monthly historical report that goes into a ticket or a management summary.

Shipping Access Logs to Object Storage and Running in Containers

The operational details — how to ship access logs to S3 or Azure Blob with Fluent Bit, Vector, or Filebeat/Logstash, how to handle concurrent writes and log loss at high traffic, and how to run AWFFull as a stateless container against object storage — are covered in the companion post:

AWFFull in the Cloud: Shipping Logs to Object Storage and Running in Containers

The Big Ticket Items

PCRE → PCRE2

AWFFull used libpcre (PCRE version 1) for log-line regex parsing. PCRE 1 has been officially end-of-life since 2012 and is being dropped by distributions. We migrated to **PCRE2**, the current maintained successor with a cleaner API and better Unicode support. The change touched `parser.c`, `common.h`, and `configure.ac`.

GeoIP: From Legacy `.dat` to libmaxminddb

The old code used **libGeoIP** with a proprietary binary `.dat` format that MaxMind deprecated in 2019. Country lookups were broken for anyone who hadn't kept a seven-year-old database around.

AWFFull 4.0.0 replaces this with **libmaxminddb**, the open standard MMDB format used by both MaxMind (GeoLite2) and DB-IP. The migration is transparent — IPv4 and IPv6 are both handled by a single `MMDB_lookup_string()` call, with no separate code paths needed.

Free databases updated monthly:

- **Country**: DB-IP Country Lite
- **ASN**: DB-IP ASN Lite

ASN Statistics: New Table and Pie Chart

Once you have a MMDB-capable lookup layer, adding ASN (Autonomous System Number) statistics is almost free. A second database (`GeoASNDatabase`) maps IPs to their network operator —

think “Cloudflare”, “Amazon AWS”, “Deutsche Telekom”. The report now shows a **Top ASNs** table and pie chart, answering the question *“which networks send the most traffic to my site?”*

```
GeoASN          yes
GeoASNDatabase  /usr/share/GeoIP/dbip-asn-lite.mmdb
```

Performance note: both GeoIP and GeoASN lookups happen in the *output phase*, once per unique IP address — not once per log line. On a typical site the overhead is negligible.

Hyperscan: SIMD Pattern Matching

AWFFull supports extensive pattern lists for ignoring, grouping, and hiding hosts, URLs, referrers, and agents. The original code used a **Boyer-Moore-Horspool** linear scan: each log record was matched against every pattern in turn.

When **Intel Hyperscan** (`libhyperscan`) is installed, AWFFull now compiles all patterns from a given list into a single Hyperscan database and matches them in one SIMD pass per record. For large pattern lists — say, 50+ bot/spider agent patterns — this is significantly faster. Hyperscan is auto-detected at configure time and falls back to the existing BMH path if unavailable.

```
sudo apt install libhyperscan-dev
./configure && make
```

mimalloc: Better Heap Performance

For large log files with millions of unique hosts, URLs, and referrers, heap allocation overhead becomes measurable. Adding **mimalloc** (Microsoft’s high-performance allocator) as an optional drop-in was straightforward via the explicit `mi_calloc` API. Auto-detected at configure time.

XXH3 Hash Function

The internal hash table used djb2, a good general-purpose hash from 1991. AWFFull 4.0.0 replaces it with **XXH3** (via the bundled `xxhash.h`), which offers better distribution and significantly higher throughput on modern CPUs. The table load-factor was also tuned to reduce collisions.

User-Visible Improvements

Date Stamp on Graphs

Every generated PNG graph now shows the **current date** (`YYYY-MM-DD`) in the top-right corner, aligned with the title baseline. Small but useful when you’re looking at archived reports six months later.

Next / Previous Month Navigation

Monthly report pages now show navigation links at the top:

```
← February 2026  |  [Index]  |  April 2026 →
```

Links are derived from the history file — only months that actually exist in the database are linked.

Updated Browser/Bot Detection

The `sample.conf` browser/agent detection list dated from 2005 and contained entries like "Netscape 4.5", "Internet Explorer 3.0 (Win95)", and "MSNBot" — none of which have been seen in the wild for a decade or more. The list has been completely rewritten:

Removed: Netscape 1–4.x, Internet Explorer 3–7, Firebird, Galeon, Camino, msnbot (all entries)

Added: Chrome/Chromium, Edge (Chromium), Brave, Samsung Browser, Firefox, Safari, Opera (OPR/), modern search bots (Bingbot, DuckDuckBot, YandexBot, Baidu, Applebot, AhrefsBot, SemrushBot), AI crawlers (GPTBot, ClaudeBot, PerplexityBot, Bytespider, CCBot, Amazonbot), and CLI tools (curl, Wget, python-requests).

Build System and Tooling

CLI Cleanup: 14 Flags Removed

AWFFull inherited many single-letter flags from the original Webalizer. Flags like `-G` (suppress hourly graph), `-L` (suppress legend), `-M` (mangle agents), `-x` (HTML extension), and `-I` (index alias) had exact equivalents in the config file. At version 4.0.0 — a major version bump — these were removed from the CLI, along with three fully deprecated no-op flags (`-d`, `-q`, `-Q`).

The `--help` output now documents the new verbosity levels (`-v` through `-vvvvv`) and includes a note pointing config-file-only settings to `awffull.conf(5)`.

Man Pages: DocBook XML → Pandoc Markdown

The existing man pages were written in DocBook XML 4.5 and built with `xsltproc` — a tool that requires `xsltproc`, `docbook-xsl`, and several XML schema packages, none of which are installed by default on most systems in 2026.

The man pages are now maintained as **pandoc Markdown** (`doc/awffull.1.md`, `doc/awffull.conf.5.md`) and built with:

```
pandoc -s -t man awffull.1.md -o awffull.1
```

`configure.ac` auto-detects pandoc. Pre-built man pages remain in the repository as a fallback for systems without pandoc. The DocBook XML sources are retained in `doc/` for historical reference.

Translations: Quality Over Quantity

AWFFull shipped with 33 language files, most of which were approximately 29% complete and had not been updated since 2008. Rather than shipping incomplete translations that silently fall back to English mid-report, we reduced to **6 fully complete languages** (German, Finnish, Italian, Swedish, Brazilian Portuguese, Indonesian) and brought all of them to **100% (524/524 strings)**.

The `.pot` template was regenerated with `xgettext` to include all new strings added in 4.0.0 (GeoASN messages, MaxMind DB errors, pattern matching warnings).

Repository Modernisation

The project moved from a collection of legacy files to a cleaner structure:

Before	After
`README` + `README.FIRST` + `README.webalizer`	`README.md` (single entry point)
`ChangeLog` (GNU format, frozen)	`CHANGELOG.md` (Keep a Changelog)
`COPYING`	`LICENSE`
`TODO` (plain text, 2008)	`TODO.md` (Markdown with `[x]` checkboxes)
`country-codes.txt` (2005, informational)	removed
`DNS.README` (dead link)	removed
`contrib/` (Webalizer migration scripts)	removed
`config.guess` from 2008	updated to automake 1.16 (2022)

> **tip**
>
> The full list of technical changes is available in CHANGES.md.

CI/CD Pipeline

A full GitLab CI pipeline was added covering:

- **Build verification** on Debian 12/13, Ubuntu 22.04/24.04, AlmaLinux 8/9/10
- **Source and binary tarballs** on every push to `main`
- **`.deb` package** (Ubuntu 24.04 amd64) on release tags
- **`.rpm` package** (AlmaLinux 9 amd64) on release tags
- **Container image** pushed to GitLab Container Registry (Ubuntu 24.04 base, all optional libs included)
- **GitLab Release** creation with all artifacts uploaded to the Generic Package Registry

Releases are triggered by pushing a semver tag:

```
git tag v4.0.0
git push --tags
```

The container image is available as:

```
docker pull registry.gitlab.com/aleks001/awffull:v4.0.0
docker pull registry.gitlab.com/aleks001/awffull:latest
```

The other artifacts are available on the Release Page

What's Next

The `TODO.md` tracks longer-term ideas: SVG graphs (replacing the aging libgd dependency), weekly summaries, a templating system for the HTML output, and better browser/OS reporting. Contributions welcome.

The project is at https://gitlab.com/aleks001/awfull.

AWFFull is free software licensed under GPL v3 or later. Based on The Webalizer by Bradford L. Barrett (1997–2001). Extended by Steve McInerney (2004–2008) and Aleksandar Lazic (2026–present).