

Argus

Counts the attacks against your servers and keeps the history as charts, with no permanent service and no database

Manual

Version 1.0.0-14
Commit 9622e76
Commit date 3 November 2026
Licence MIT

Offline manual, identical to the guides of the repository at the version and commit shown. The text of the repository is the reference.

Contents

Overview **1**
- Install in two steps . . . 1
- The three views . . . 1
- One connection per host, no retry . . . 2
- Configuration . . . 2
- Check that it works . . . 2
- Documentation . . . 2
- Credits . . . 2
- License . . . 3

Install Argus on the server **4**
- What you get . . . 4
- Requirements . . . 4
- Install . . . 4
- Try it now . . . 4
- Change a setting . . . 5
- Remove . . . 5

Copy the charts to your desk and read them **6**
- What you get . . . 6
- Requirements . . . 6
- Declare the servers . . . 6
- Copy . . . 6
- One connection, no retry . . . 6
- Read the charts . . . 7

Charts: windows, split, hour, series **8**
- Windows . . . 8
- Split a long window . . . 8
- The hour of the day . . . 8
- Series . . . 9
- Read the numbers instead of the picture . . . 9
- Slices at hour 00 (`T00`) . . . 9
- Chart language . . . 9

Troubleshooting **10**
- Run the check . . . 10
- What each line means . . . 10
- Run a collector by hand . . . 10
- Common causes . . . 10

Overview

No permanent service, no database, suited to machines with less than 1 GB of RAM.

Argus counts the attacks against your servers, SSH by itself and, when they are installed, the bans of Fail2Ban and CrowdSec, and keeps the history as charts. A nightly cron job writes plain text files and draws the charts. Nothing runs between two nights.

Argus never stores or shows an attacker's IP address, only counts per hour and per day. An **attack** is a source IP that reaches a minimum number of failed authentication attempts within one day (5 by default), so one mistyped password does not count.

Grafana is a continuous view of an infrastructure: a service that runs all the time and can be queried live. Argus is a periodic snapshot of your security history: one image per night, read when you want. Grafana can aggregate by week or by year too, at the price of a permanent service, a time-series database and a dashboard to build; Argus has none of the three, which is why it fits a 1 GB server.

Install in two steps

If you usually install a program on the computer you work on, here it is the reverse. The package goes on the **server**, the machine that is attacked. Your own computer, the desk, installs nothing of the package and only receives the images.

1. **On the server** (the machine that is attacked), install the package. It counts the attacks and draws the charts every night. There is nothing to launch afterwards.
2. **On your desk** (the computer you read from), run `argus-fetch`. It copies the chart images from your servers to your computer.

Where each part lives:

```
server.example.org (the server)          your desk
─────────────────────────────────────    ──────────────────────────────
/opt/argus/        scripts               ~/.config/argus/hosts.conf
/etc/argus/        argus.conf                the list of your servers
/srv/argus/<year>/ data and images       ~/argus-backup/<server>/<year>/images/
/usr/bin/argus-chart                         the copied charts
/usr/bin/argus-fetch   ── copies images ──►
```

Step 1, on the server (Debian or Ubuntu):

```
sudo apt install ./argus_1.0.0-14_all.deb
```

Step 2, on your desk. Put your servers in the file `~/.config/argus/hosts.conf`, one per line, as `<ssh alias>,<data directory>`, then run `argus-fetch`:

```
mkdir -p ~/.config/argus
printf 'server.example.org,/srv/argus/\n' > ~/.config/argus/hosts.conf
argus-fetch
```

`argus-fetch` is installed by the same package. On a desk that is not a Debian machine, copy the single script `argus-fetch` from this repository; it needs only `bash`, `ssh` and `rsync`. Details: `docs/install.md` and `docs/desk.md`.

The three views

`argus-chart` draws a chart for the window you choose. Run it on the server. Each command writes a PNG in `/srv/argus/<year>/images/` (add `--out DIR` to write elsewhere, which an ordinary user needs).

- `argus-chart daily` is the history of the year, one point per day.
- `argus-chart period` is the count of each series over a window.

- `argus-chart hourly` is the count per hour of the day over a window, from 00 to 23.

Ten windows to copy:

```
argus-chart period --day 2026-10-30                     # one day, one point per hour
argus-chart period --week 2026-W44                      # ISO week, Monday to Sunday
argus-chart period --month 2026-10                      # one calendar month
argus-chart period --quarter 2026-Q4                    # one quarter
argus-chart period --year 2025                          # one year
argus-chart period --from 2025-12-01 --to 2026-01-31    # a range over two years
argus-chart period --month 2026-10 --split week         # one PNG per week of the month
argus-chart period --week 2026-W44 --series ssh_direct  # one series only
argus-chart hourly --week 2026-W44                      # which hours are attacked
argus-chart hourly --month 2026-10 --series ssh_direct,ip_ban_sshd
```

Series are drawn side by side, never added up: SSH, each Fail2Ban jail and each CrowdSec scenario keep their own line and colour. Details: `docs/charts.md`.

One connection per host, no retry

`argus-fetch` opens **one connection per host** and **never retries**. Many servers ban an address after a few failed logins, so a second attempt could lock you out. A host that does not answer is reported once, the other hosts are still copied, and the exit code is 1. Fix the cause (key, alias, firewall, machine off), then run it again by hand.

Configuration

Everything works with the defaults. To change one, edit `/etc/argus/argus.conf` on the server; the file is read by every script and ships with each value commented out:

```
# /etc/argus/argus.conf
ARGUS_SSH_THRESHOLD=5          # failed attempts of one IP in one day that make an attack
ARGUS_LANG=en                  # chart language: en, fr, de or es
ARGUS_DATA_DIR="/srv/argus"    # where the data and the images are written
```

The other variables (log paths, image size, series colours) are listed in `man argus`.

Check that it works

```
sudo /opt/argus/argus-doctor.sh
```

It tells whether each source (SSH, Fail2Ban, CrowdSec) was collected yesterday. See `docs/troubleshooting.md`.

Documentation

- `docs/install.md`: install on the server, layout, uninstall
- `docs/desk.md`: copy the charts to your desk and read them
- `docs/charts.md`: windows, split, hour, series, time zone, `T00` slices
- `docs/troubleshooting.md`: what `argus-doctor.sh` says and what to do
- Manual pages: `man argus`, `man argus-chart`, `man argus-fetch` (sources in `man/en/`)

Credits

Argus is the result of a human-LLM collaboration, with the roles written down as they were actually played. The full list is in `CONTRIBUTORS.md`.

- **Ronan Davalan** — Architect and arbiter: original scripts (2024), decisions, validation and testing.

- **Claude Code (Anthropic)** — Lead developer: collectors, chart rendering, `argus-doctor.sh`, Debian package, documentation.
- **Gemini (Google)** — Synthesizer and strategic co-writer: planning and framing of the project's purpose.
- **Gemini (Google), GLM-5.3-Flash Free and Gemini 4 Argon** — Graphic designers: the project's visual identity (mark, icon, favicon), adopted by Ronan Davalan on 2026-11-02.
- **DeepSeek** — Field companion: CrowdSec setup on a real server used to validate `collect-crowdsec.sh`.
- **DeepSeek Flash** — Site content in English, then the French, German and Spanish translations of the documentation, manual pages and site.
- **Muse Spark 1.3 Contributor** — Site styling: integration of the logo, the favicon and the chart image.
- **gd2:** Ronan Davalan's late brother. He taught Ronan Git and how to use it, advised on this project, and made the first outside edit to its README (2024-08-07).

License

MIT — see `LICENSE`.

Install Argus on the server

The package goes on the **server**, the machine whose logs Argus reads. You do not install it on your desk: the desk only copies the charts (see `desk.md`).

What you get

```
/opt/argus/
├── collect-ssh.sh         SSH attacks, from auth.log or journald
├── collect-fail2ban.sh    Fail2Ban bans, one series per jail (optional)
├── collect-crowdsec.sh    CrowdSec alerts, one series per scenario (optional)
├── chart-daily.sh         history of the year, one point per day
├── chart-period.sh        count over a window (day, week, month, quarter, year, range)
├── chart-hourly.sh        count per hour of the day over a window
├── chart-*.gp             the gnuplot drawing of each view
├── argus-lib.sh           code shared by the chart scripts
└── argus-doctor.sh        health check of the collection

/usr/bin/argus-chart       one command for the three views
/usr/bin/argus-fetch       copies the charts to a desk
/etc/argus/argus.conf      optional settings, every line commented out
/etc/cron.d/argus          the nightly jobs
/srv/argus/                data and charts, readable by everyone
└── 2026/
    ├── ssh_direct_2026.txt      one line per hour with an attack
    ├── ip_ban_sshd_2026.txt     one file per Fail2Ban jail, if present
    ├── .daily/                  daily totals, rebuilt by chart-daily.sh
    ├── .heartbeat/              daily proof that each collector ran
    └── images/                  the PNG charts
```

`/srv/argus/` holds only counts, never an IP address, so it is readable by every user: you copy the charts without `sudo`.

Requirements

A Debian or Ubuntu server with cron. The package depends on `gawk` and `gnuplot-nox`; `apt` installs them. `rsync` and `openssh-client` are recommended (they serve `argus-fetch`). Fail2Ban, CrowdSec and `jq` are optional. A server with 1 GB of RAM is enough: nothing stays running.

Install

Download `argus_1.0.0-14_all.deb` from the releases page of the repository, then, on the server:

```
sudo apt install ./argus_1.0.0-14_all.deb
```

The cron jobs start that night: collection at 00:05, charts at 00:15, health check at 00:20. The first chart appears in `/srv/argus/2026/images/` after the first night.

Try it now

To skip the first night, run the steps by hand on a server that already has log history:

```
sudo /opt/argus/collect-ssh.sh
sudo /opt/argus/chart-daily.sh
ls /srv/argus/2026/images/
```

Then draw a view:

```
sudo argus-chart period --month 2026-10
```

Change a setting

`/etc/argus/argus.conf` is a shell file read by every script. Uncomment and edit only what you need:

```
# /etc/argus/argus.conf
ARGUS_SSH_THRESHOLD=10         # failed attempts of one IP in one day that make an attack
ARGUS_LANG=en                  # chart language: en, fr, de or es
ARGUS_IMG_WIDTH=3840           # image size in pixels
ARGUS_IMG_HEIGHT=2160
ARGUS_COULEUR_ssh_direct=orange   # colour of one series, any gnuplot colour name
```

Variable names use capitals and underscores, never hyphens. The series `caddy-xmlrpc` is therefore coloured by `ARGUS_COULEUR_caddy_xmlrpc`. All variables are listed in `man argus`.

Remove

```
sudo apt remove argus     # removes the scripts, keeps /srv/argus and /etc/argus
sudo apt purge argus      # also removes /srv/argus and /etc/argus
```

Purge deletes your history. Copy it first with `argus-fetch` (see `desk.md`).

Copy the charts to your desk and read them

Argus runs on your servers and draws the charts there. `argus-fetch` copies the images to the computer you work on. Nothing is sent from the server: you pull when you want.

What you get

```
~/.config/argus/hosts.conf              the servers to copy, one per line
~/argus-backup/
└── server.example.org/                 one folder per ssh alias
    └── 2026/
        └── images/
            ├── 2026-11-02_argus_4k.png
            └── 2026-11-02_argus_period_week-2026-W44.png
```

Requirements

`bash`, `ssh` and `rsync` on the desk. The package installs `argus-fetch` in `/usr/bin`; on another system, copy the one file `argus-fetch` from the repository to a folder of your `PATH`. Your `ssh` must already log in to the server with a key, without a password prompt: `argus-fetch` never asks for one.

Declare the servers

One server per line, `<ssh alias>,<data directory>`. The alias is a `Host` name of `~/.ssh/config` (or `user@host`); the directory is `ARGUS_DATA_DIR` on that server, `/srv/argus/` by default.

```
mkdir -p ~/.config/argus
printf 'server.example.org,/srv/argus/\nbackup.example.org,/srv/argus/\n' > ~/.config/argus/hosts.conf
cat ~/.config/argus/hosts.conf
```

Empty lines and lines starting with `#` are ignored. The repository ships `hosts.conf.example`.

Copy

```
argus-fetch
```

For each server it opens **one** connection, copies the PNG files of `<directory>/<year>/images/`, and closes. A file already on your desk is never overwritten and nothing is deleted. Run it as often as you like: only new images are copied.

To copy one server only, or to put the copies elsewhere:

```
argus-fetch --only server.example.org
argus-fetch --dest ~/Documents/argus
```

One connection, no retry

If a server does not answer, `argus-fetch` says so once on the error output, goes on with the other servers, and ends with the exit code 1. It never tries again, because many servers ban an address after a few failed logins. Find the cause (key, alias, firewall, server off), then run the command again yourself.

Read the charts

Open the images with any viewer:

```
xdg-open ~/argus-backup/server.example.org/2026/images/
```

A chart shows one line per series. On a window of up to 31 points, the exact count is written at each point. A view that does not exist yet (another week, a month by hour) is drawn on the server, as root so that the image lands in the images folder:

```
ssh server.example.org
sudo argus-chart period --week 2026-W44
```

The next `argus-fetch` copies the new image with the others. See `charts.md` for the windows you can ask for.

Charts: windows, split, hour, series

`argus-chart` has three views. Run it on the server, as root to write in `/srv/argus/<year>/images/`, or as an ordinary user with `--out DIR`.

```
argus-chart daily  [YEAR]              one point per day, the whole year
argus-chart period [window] [options]  one line per series over a window
argus-chart hourly [window] [options]  24 positions, 00 to 23, per series
```

Each file is named `<today>_argus_<view>_<window>.png`, for example `2026-11-02_argus_period_week-2026-W44.png`.

Windows

`period` and `hourly` take the same windows. Without one, they use the current year.

Option	Window	Example
`--day YYYY-MM-DD`	one day	`--day 2026-10-30`
`--week YYYY-Www`	ISO week, Monday to Sunday	`--week 2026-W44`
`--month YYYY-MM`	calendar month	`--month 2026-10`
`--quarter YYYY-Qn`	quarter, `n` from 1 to 4	`--quarter 2026-Q4`
`--year YYYY`	calendar year	`--year 2025`
`--from D --to D`	range, both days included	`--from 2025-12-01 --to 2026-01-31`

Weeks follow ISO 8601: week 1 of 2027 starts on Monday 2027-01-04, and 2026 has a week 53. A window that spans two years reads the data of both.

```
argus-chart period --day 2026-10-30 --out ~/argus-out
argus-chart period --from 2025-12-01 --to 2026-01-31 --out ~/argus-out
```

Split a long window

`--split` draws one chart per sub-period instead of one long chart:

```
argus-chart period --month 2026-10 --split week --out ~/argus-out
argus-chart period --year 2026 --split month --out ~/argus-out
```

`--split week` suits a month or a quarter, `--split month` a quarter or a year. The first command writes one file per ISO week touching October, each cut to the window, named `…_month-2026-10_week-2026-W41.png`.

The hour of the day

`hourly` adds up, for each series, the attacks seen at each hour over the window, and draws 24 positions. It shows at which hours of the day you are attacked, not on which days.

```
argus-chart hourly --week 2026-W44 --out ~/argus-out
```

The hour is the **local time of the server**, as written by the collectors. It is the hour at which the attack was seen in the logs, not the hour it began. The title says "local server time". A change of daylight saving time adds or removes one hour on one day; over a week or more the effect does not show.

Series

A series is one line: `ssh_direct` for SSH, `ip_ban_<jail>` for each Fail2Ban jail, `ip_ban_<scenario>` for each CrowdSec scenario. The name is the data file name without the year, so `ssh_direct_2026.txt` is the series `ssh_direct`.

Series are overlaid, never added up: Argus does not invent one score from different tools. Each series keeps the same colour from year to year (set it with `ARGUS_COULEUR_<series>`, otherwise it comes from a hash of the name).

```
argus-chart period --week 2026-W44 --list-series
argus-chart period --week 2026-W44 --series ssh_direct
argus-chart hourly --month 2026-10 --series ssh_direct,ip_ban_sshd --out ~/argus-out
```

An unknown name in `--series` stops the command with the exit code 2 and lists the available names.

Read the numbers instead of the picture

`--print-data` prints the table behind the chart on standard output (tab-separated, header starting with `#`) and draws nothing:

```
argus-chart period --week 2026-W44 --print-data
argus-chart hourly --day 2026-10-30 --print-data
```

Slices at hour 00 (`T00`)

A data line is `<count> <date>T<hour>`, for example `492 2026-10-30T14`. Files from before the collectors were hourly hold one line per day, all written as `T00`. A series made only of `T00` slices has no hour information, so `hourly` leaves it out instead of drawing one bar at midnight. It prints the message `series 'legacy' excluded: all its slices are at hour 00 (daily data, no hourly depth)` on the error output and writes `excluded series: no hourly data: legacy` in the title of the chart. `period` and `daily` still draw it, as daily points.

Chart language

`ARGUS_LANG` sets the titles and the axes: `en` (default), `fr`, `de` or `es`.

```
ARGUS_LANG=de argus-chart period --week 2026-W44 --out ~/argus-out
```

Troubleshooting

A chart with no attack can mean a quiet day or a collector that stopped. The data files cannot tell them apart, because a day without attack writes no line. `argus-doctor.sh` can: every collector writes one line a day in `/srv/argus/<year>/.heartbeat/<source>.txt`, `<date> <number of events>`, even when the number is 0.

Run the check

```
sudo /opt/argus/argus-doctor.sh
```

The cron job runs it every night at 00:20 and appends the output to `/var/log/argus.log`. The exit code is 0 when there is nothing to report, 1 otherwise. A healthy server prints:

```
== Source availability ==

== Heartbeats (2026) ==
  ssh : ok
    -> up to date, yesterday: 12 event(s)
  fail2ban : absent
  crowdsec : absent

Nothing to report.
```

What each line means

`<source> : ok` or `absent` says whether the source can be read now: `auth.log` or `journald` for `ssh`, `fail2ban-client` and its log for `fail2ban`, `cscli` and `jq` for `crowdsec`. `absent` for Fail2Ban or CrowdSec is normal when you do not run them.

Line under the source	Meaning	What to do
`up to date, yesterday: N event(s)`	collected yesterday, N events (0 is a quiet day)	nothing
`last confirmed on DATE (no entry for yesterday)`	the collector did not run or failed since DATE	look at `/var/log/argus.log`, run the collector by hand
`never collected`	no heartbeat file, though the source is readable	run the collector by hand, check `/etc/cron.d/argus`

Lines that need action are repeated under `TO CHECK` on the error output.

Run a collector by hand

```
sudo /opt/argus/collect-ssh.sh
sudo /opt/argus/argus-doctor.sh
tail -n 20 /var/log/argus.log
```

Common causes

- **No image after the first night.** Check that `/etc/cron.d/argus` exists and that cron runs (`systemctl status cron`). Run `sudo /opt/argus/chart-daily.sh` and read its message.

- **`chart-daily.sh: no data file for 2026 in /srv/argus/2026 — nothing to plot.`** No collector has written a file this year. A server without any attack writes none. Run `sudo /opt/argus/collect-ssh.sh` to collect yesterday.
- **SSH shows 0 events every day.** `auth.log` is absent and `journald` holds the logs under another unit name. Set `ARGUS_SSHD_UNIT="sshd"` in `/etc/argus/argus.conf` (the default is `ssh`).
- **Fewer SSH attacks than Fail2Ban bans.** Argus counts an IP only after `ARGUS_SSH_THRESHOLD` failed attempts in one day (5 by default), and Fail2Ban can ban earlier. Lower the threshold to compare.
- **`argus-chart: unknown view.`** The views are `period`, `hourly` and `daily`.
- **`unknown series.`** List the names with `argus-chart period --list-series`.
- **`argus-fetch` reports a host.** The server did not answer on the first try, and `argus-fetch` does not retry. Check the alias, the key and the firewall, then run it again. See `desk.md`.
- **Permission denied writing an image.** `/srv/argus` belongs to root: use `sudo`, or `--out DIR` in your home.