

Argus

Cuenta los ataques contra sus servidores y guarda el historial en gráficos, sin servicio permanente ni base de datos

Manual

Versión	1.0.0-14
Commit	9622e76
Fecha del commit	3 de noviembre de 2026
Licencia	MIT

Manual sin conexión, idéntico a las guías del repositorio en la versión y el commit indicados. El texto de referencia es el del repositorio.

Índice

Resumen **1**
- Instalación en dos tiempos . . . 1
- Las tres vistas . . . 1
- Una conexión por host, sin reintento . . . 2
- Configuración . . . 2
- Comprobar que funciona . . . 2
- Documentación . . . 2
- Créditos . . . 2
- Licencia . . . 3

Instalar Argus en el servidor **4**
- Lo que obtiene . . . 4
- Requisitos . . . 4
- Instalación . . . 4
- Probarlo ahora . . . 4
- Cambiar un ajuste . . . 5
- Desinstalación . . . 5

Traer los gráficos al equipo de escritorio y leerlos **6**
- Lo que obtiene . . . 6
- Requisitos . . . 6
- Declarar los servidores . . . 6
- Copiar . . . 6
- Una conexión, sin reintento . . . 6
- Leer los gráficos . . . 7

Gráficos: ventanas, división, hora, series **8**
- Ventanas . . . 8
- Dividir una ventana larga . . . 8
- La hora del día . . . 8
- Las series se superponen, nunca se suman . . . 9
- Leer los números en lugar de la imagen . . . 9
- Franjas a la hora 00 (`T00`) . . . 9
- Idioma de los gráficos . . . 9

Solución de problemas **10**
- Ejecutar la comprobación . . . 10
- Qué significa cada línea . . . 10
- Ejecutar un recolector a mano . . . 10
- Causas frecuentes . . . 10

Resumen

Sin servicio permanente, sin base de datos, adecuado para máquinas con menos de 1 GB de RAM.

Argus cuenta los ataques contra sus servidores, SSH por sí solo y, cuando están instalados, los bloqueos de Fail2Ban y CrowdSec, y conserva el historial como gráficos. Una tarea cron se ejecuta cada noche, escribe archivos de texto plano y dibuja los gráficos. Entre dos noches no se ejecuta nada.

Argus nunca almacena ni muestra la dirección IP de un atacante, solo recuentos por hora y por día. Un **ataque** es una IP de origen que alcanza un número mínimo de intentos de autenticación fallidos en un día (5 por defecto), de modo que una contraseña mal escrita no cuenta.

Grafana es una vista continua de una infraestructura: un servicio que se ejecuta todo el tiempo y puede consultarse en directo. Argus es una instantánea periódica de su historial de seguridad: una imagen por noche, leída cuando usted quiera. Grafana también puede agregar por semana o por año, a costa de un servicio permanente, una base de datos de series temporales y un panel que construir; Argus no tiene ninguno de los tres, por eso cabe en un servidor de 1 GB.

Instalación en dos tiempos

Si normalmente instala un programa en el ordenador en el que trabaja, aquí es al revés. El paquete va en el **servidor**, la máquina atacada. Su propio ordenador, el equipo de escritorio, no instala nada del paquete y solo recibe las imágenes.

1. **En el servidor** (la máquina atacada), instale el paquete. Cuenta los ataques y dibuja los gráficos cada noche. Después no hay nada que lanzar.
2. **En su equipo de escritorio** (el ordenador desde el que lee), ejecute `argus-fetch`. Copia las imágenes de los gráficos de sus servidores a su ordenador.

Dónde está cada parte:

```
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 ──►
```

Paso 1, en el servidor (Debian o Ubuntu):

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

Paso 2, en su equipo de escritorio. Ponga sus servidores en el archivo `~/.config/argus/hosts.conf`, uno por línea, con la forma `<ssh alias>,<data directory>`, y luego ejecute `argus-fetch`:

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

`argus-fetch` lo instala el mismo paquete. En un equipo que no sea una máquina Debian, copie el único script `argus-fetch` desde este repositorio; solo necesita `bash`, `ssh` y `rsync`. Detalles: `docs/install.md` y `docs/desk.md`.

Las tres vistas

`argus-chart` dibuja un gráfico para la ventana que usted elija. Ejecútelo en el servidor. Cada comando escribe un PNG en `/srv/argus/<year>/images/` (añada `--out DIR` para escribir en otro sitio, lo que necesita un usuario normal).

- `argus-chart daily` es el historial del año, un punto por día.
- `argus-chart period` es el recuento de cada serie en una ventana.
- `argus-chart hourly` es el recuento por hora del día en una ventana, de 00 a 23.

Diez ventanas que copiar:

```
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
```

Las series se dibujan una al lado de otra, nunca sumadas: SSH, cada jail de Fail2Ban y cada escenario de CrowdSec conservan su propia línea y su propio color. Detalles: `docs/charts.md`.

Una conexión por host, sin reintento

`argus-fetch` abre **una conexión por host** y **nunca reintenta**. Muchos servidores bloquean una dirección tras unos intentos de conexión fallidos, así que un segundo intento podría dejarle sin acceso. Un host que no responde se notifica una vez, los demás hosts se copian igualmente y el código de salida es 1. Corrija la causa (clave, alias, cortafuegos, máquina apagada) y vuelva a ejecutarlo a mano.

Configuración

Todo funciona con los valores por defecto. Para cambiar uno, edite `/etc/argus/argus.conf` en el servidor; el archivo lo lee cada script y se entrega con cada valor comentado:

```
# /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
```

Las demás variables (rutas de registros, tamaño de imagen, colores de las series) se enumeran en `man argus`.

Comprobar que funciona

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

Indica si cada fuente (SSH, Fail2Ban, CrowdSec) se recogió ayer. Véase `docs/troubleshooting.md`.

Documentación

- `docs/install.md`: instalar en el servidor, estructura, desinstalación
- `docs/desk.md`: traer los gráficos al equipo de escritorio y leerlos
- `docs/charts.md`: ventanas, división, hora, series, zona horaria, franjas `T00`
- `docs/troubleshooting.md`: qué dice `argus-doctor.sh` y qué hacer
- Páginas de manual: `man argus`, `man argus-chart`, `man argus-fetch` (fuentes en `man/en/`)

Créditos

Argus es el resultado de una colaboración entre un humano y varios modelos de lenguaje, con los roles escritos tal como se desempeñaron realmente. La lista completa está en `CONTRIBUTORS.md`.

- **Ronan Davalan** — Arquitecto y árbitro: scripts originales (2024), decisiones, validación y pruebas.
- **Claude Code (Anthropic)** — Desarrollador principal: recolectores, renderizado de gráficos, `argus-doctor.sh`, paquete Debian, documentación.
- **Gemini (Google)** — Sintetizador y coautor estratégico: planificación y encuadre del propósito del proyecto.
- **Gemini (Google), GLM-5.3-Flash Free y Gemini 4 Argon** — Diseñadores gráficos: identidad visual del proyecto (marca, icono, favicon), adoptada por Ronan Davalan el 2026-11-02.
- **DeepSeek** — Compañero de campo: instalación de CrowdSec en un servidor real que sirvió para validar `collect-crowdsec.sh`.
- **DeepSeek Flash** — Contenido del sitio en inglés, luego traducciones al francés, alemán y español de la documentación, las páginas de manual y el sitio.
- **Muse Spark 1.3 Contributor** — Diseño del sitio: integración del logo, del favicon y de la imagen del gráfico.
- **gd2:** hermano fallecido de Ronan Davalan. Le enseñó Git a Ronan y su uso, asesoró en este proyecto e hizo la primera modificación externa de su README (2024-08-07).

Licencia

MIT — véase `LICENSE`.

Instalar Argus en el servidor

El paquete va en el **servidor**, la máquina cuyos registros lee Argus. No lo instale en su equipo de escritorio: el equipo de escritorio solo copia los gráficos (véase `desk.md`).

Lo que obtiene

```
/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/` solo contiene recuentos, nunca una dirección IP, por lo que cualquier usuario puede leerlo: usted copia los gráficos sin `sudo`.

Requisitos

Un servidor Debian o Ubuntu con cron. El paquete depende de `gawk` y `gnuplot-nox`; `apt` los instala. Se recomiendan `rsync` y `openssh-client` (sirven a `argus-fetch`). Fail2Ban, CrowdSec y `jq` son opcionales. Basta un servidor con 1 GB de RAM: nada queda en ejecución.

Instalación

Descargue `argus_1.0.0-14_all.deb` de la página de versiones del repositorio y luego, en el servidor:

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

Las tareas cron empiezan esa noche: recogida a las 00:05, gráficos a las 00:15, comprobación de estado a las 00:20. El primer gráfico aparece en `/srv/argus/2026/images/` tras la primera noche.

Probarlo ahora

Para saltarse la primera noche, ejecute los pasos a mano en un servidor que ya tenga historial de registros:

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

Luego dibuje una vista:

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

Cambiar un ajuste

`/etc/argus/argus.conf` es un archivo de shell que lee cada script. Descomente y modifique solo lo que necesite:

```
# /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
```

Los nombres de variables usan mayúsculas y guiones bajos, nunca guiones. Por eso la serie `caddy-xmlrpc` se colorea con `ARGUS_COULEUR_caddy_xmlrpc`. Todas las variables se enumeran en `man argus`.

Desinstalación

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

La purga también borra su historial. Cópielo primero con `argus-fetch` (véase `desk.md`).

Traer los gráficos al equipo de escritorio y leerlos

Argus se ejecuta en sus servidores y dibuja los gráficos allí. `argus-fetch` copia las imágenes al ordenador en el que trabaja. No se envía nada desde el servidor: usted trae las imágenes cuando quiere.

Lo que obtiene

```
~/.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
```

Requisitos

`bash`, `ssh` y `rsync` en el equipo de escritorio. El paquete instala `argus-fetch` en `/usr/bin`; en otro sistema, copie el único archivo `argus-fetch` del repositorio a una carpeta de su `PATH`. Su `ssh` ya debe iniciar sesión en el servidor con una clave, sin pedir contraseña: `argus-fetch` nunca la pide.

Declarar los servidores

Un servidor por línea, `<ssh alias>,<data directory>`. El alias es un nombre de `Host` de `~/.ssh/config` (o `user@host`); el directorio es `ARGUS_DATA_DIR` en ese servidor, `/srv/argus/` por defecto.

```
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
```

Las líneas vacías y las líneas que empiezan por `#` se ignoran. El repositorio incluye `hosts.conf.example`.

Copiar

```
argus-fetch
```

Para cada servidor abre **una** conexión, copia los archivos PNG de `<directory>/<year>/images/` y la cierra. Un archivo que ya esté en su equipo de escritorio nunca se sobrescribe y no se borra nada. Ejecútelo tantas veces como quiera: solo se copian las imágenes nuevas.

Para copiar un solo servidor o poner las copias en otro sitio:

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

Una conexión, sin reintento

Si un servidor no responde, `argus-fetch` lo indica una vez en la salida de error, continúa con los demás servidores y termina con el código de salida 1. Nunca reintenta, porque muchos servidores bloquean

una dirección tras unos intentos de conexión fallidos. Encuentre la causa (clave, alias, cortafuegos, servidor apagado) y vuelva a ejecutar el comando usted mismo.

Leer los gráficos

Abra las imágenes con cualquier visor:

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

Un gráfico muestra una línea por serie. En una ventana de hasta 31 puntos, el recuento exacto se escribe junto a cada punto. Una vista que aún no existe (otra semana, un mes por hora) se dibuja en el servidor, como root para que la imagen llegue a la carpeta de imágenes:

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

El siguiente `argus-fetch` copia la imagen nueva con las demás. Véase `charts.md` para las ventanas que puede pedir.

Gráficos: ventanas, división, hora, series

`argus-chart` tiene tres vistas. Ejecútelo en el servidor, como root para escribir en `/srv/argus/<year>/images/`, o como usuario normal con `--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
```

Cada archivo se llama `<today>_argus_<view>_<window>.png`, por ejemplo `2026-11-02_argus_period_week-2026-W44.png`.

Ventanas

`period` y `hourly` aceptan las mismas ventanas. Sin ninguna, usan el año en curso.

Opción	Ventana	Ejemplo
`--day YYYY-MM-DD`	un día	`--day 2026-10-30`
`--week YYYY-Www`	semana ISO, de lunes a domingo	`--week 2026-W44`
`--month YYYY-MM`	mes natural	`--month 2026-10`
`--quarter YYYY-Qn`	trimestre, `n` de 1 a 4	`--quarter 2026-Q4`
`--year YYYY`	año natural	`--year 2025`
`--from D --to D`	rango, ambos días incluidos	`--from 2025-12-01 --to 2026-01-31`

Las semanas siguen la norma ISO 8601: la semana 1 de 2027 empieza el lunes 2027-01-04, y 2026 tiene una semana 53. Una ventana que abarca dos años lee los datos de ambos.

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

Dividir una ventana larga

`--split` dibuja un gráfico por subperiodo en lugar de un solo gráfico largo:

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

`--split week` conviene a un mes o a un trimestre, `--split month` a un trimestre o a un año. El primer comando escribe un archivo por semana ISO que toca octubre, cada uno recortado a la ventana, llamado `…_month-2026-10_week-2026-W41.png`.

La hora del día

`hourly` suma, para cada serie, los ataques vistos a cada hora de la ventana y dibuja 24 posiciones. Muestra a qué horas del día se le ataca, no en qué días.

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

La hora es la **hora local del servidor**, tal como la escriben los recolectores. Es la hora a la que se vio el ataque en los registros, no la hora a la que empezó. El título dice «local server time». El cambio al horario de verano desplaza una hora en un día, un efecto que se difumina a lo largo de una semana completa.

Las series se superponen, nunca se suman

Cada línea de la imagen es una serie: `ssh_direct` para SSH, `ip_ban_<jail>` para cada jail de Fail2Ban, `ip_ban_<scenario>` para cada escenario de CrowdSec. Su nombre es el nombre del archivo sin el año, así que `ssh_direct_2026.txt` pertenece a `ssh_direct`.

Argus no inventa una puntuación a partir de herramientas distintas. Cada serie conserva el mismo color de un año a otro, definido con `ARGUS_COULEUR_<series>` o tomado de un hash del nombre.

```
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
```

Un nombre desconocido en `--series` detiene el comando con el código de salida 2 y enumera los nombres disponibles.

Leer los números en lugar de la imagen

`--print-data` imprime en la salida estándar la tabla que hay detrás del gráfico (separada por tabulaciones, cabecera que empieza por `#`) y no dibuja nada:

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

Franjas a la hora 00 (`T00`)

Una línea de datos es `<count> <date>T<hour>`, por ejemplo `492 2026-10-30T14`. Los archivos anteriores a la recogida por horas contienen una línea por día, todas escritas como `T00`. Una serie formada solo por franjas `T00` no tiene información de hora, así que `hourly` la excluye en lugar de dibujar una barra a medianoche. Imprime el mensaje `series 'legacy' excluded: all its slices are at hour 00 (daily data, no hourly depth)` en la salida de error y escribe `excluded series: no hourly data: legacy` en el título del gráfico. `period` y `daily` la siguen dibujando como puntos diarios.

Idioma de los gráficos

`ARGUS_LANG` define los títulos y los ejes: `en` (por defecto), `fr`, `de` o `es`.

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

Solución de problemas

Un gráfico sin ataque puede significar un día tranquilo o un recolector que se detuvo. Los archivos de datos no permiten distinguirlos, porque un día sin ataque no escribe ninguna línea. `argus-doctor.sh` sí puede: cada recolector escribe una línea al día en `/srv/argus/<year>/.heartbeat/<source>.txt`, `<date> <number of events>`, incluso cuando el número es 0.

Ejecutar la comprobación

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

La tarea cron lo ejecuta cada noche a las 00:20 y añade la salida a `/var/log/argus.log`. El código de salida es 0 cuando no hay nada que informar, 1 en caso contrario. Un servidor sano imprime:

```
== Source availability ==

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

Nothing to report.
```

Qué significa cada línea

`<source> : ok` o `absent` indica si la fuente puede leerse ahora: `auth.log` o `journald` para `ssh`, `fail2ban-client` y su registro para `fail2ban`, `cscli` y `jq` para `crowdsec`. `absent` para Fail2Ban o CrowdSec es normal cuando no los ejecuta.

Línea bajo la fuente	Significado	Qué hacer
`up to date, yesterday: N event(s)`	recogida ayer, N eventos (0 es un día tranquilo)	nada
`last confirmed on DATE (no entry for yesterday)`	el recolector no se ejecutó o falló desde DATE	mire `/var/log/argus.log`, ejecute el recolector a mano
`never collected`	no hay archivo de latido, aunque la fuente es legible	ejecute el recolector a mano, compruebe `/etc/cron.d/argus`

Las líneas que requieren acción se repiten bajo `TO CHECK` en la salida de error.

Ejecutar un recolector a mano

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

Causas frecuentes

- **Ninguna imagen tras la primera noche.** Compruebe que `/etc/cron.d/argus` existe y que cron se ejecuta (`systemctl status cron`). Ejecute `sudo /opt/argus/chart-daily.sh` y lea su mensaje.

- **`chart-daily.sh: no data file for 2026 in /srv/argus/2026 — nothing to plot.`** Ningún recolector ha escrito un archivo este año. Un servidor sin ningún ataque no escribe ninguno. Ejecute `sudo /opt/argus/collect-ssh.sh` para recoger el día de ayer.
- **SSH muestra 0 eventos cada día.** `auth.log` no existe y `journald` guarda los registros bajo otro nombre de unidad. Defina `ARGUS_SSHD_UNIT="sshd"` en `/etc/argus/argus.conf` (el valor por defecto es `ssh`).
- **Menos ataques SSH que bloqueos de Fail2Ban.** Argus cuenta una IP solo tras `ARGUS_SSH_THRESHOLD` intentos fallidos en un día (5 por defecto), y Fail2Ban puede bloquear antes. Baje el umbral para comparar.
- **`argus-chart: unknown view.`** Las vistas son `period`, `hourly` y `daily`.
- **`unknown series.`** Enumere los nombres con `argus-chart period --list-series`.
- **`argus-fetch` informa de un host.** El servidor no respondió al primer intento, y `argus-fetch` no reintenta. Compruebe el alias, la clave y el cortafuegos, y vuelva a ejecutarlo. Véase `desk.md`.
- **Permiso denegado al escribir una imagen.** `/srv/argus` pertenece a root: use `sudo`, o `--out DIR` en su directorio personal.