Filtratge antiespam

Índex

Cicle formatiu: CFGS Administració de sistemes informàtics en xarxa (ASIX)

Mòdul: 0375 - Serveis de xarxa i Internet

Sistema operatiu: Ubuntu Server 26.04 LTS

Rspamd

Rspamd és un sistema de **filtratge antispam** --- un programa que analitza cada correu entrant i decideix si és legítim, sospitós o clarament brossa, abans que arribi a la safata d'entrada de l'usuari.

Figura 1: Rspamd logo

A diferència de SPF/DKIM/DMARC (que verifiquen la **identitat** del remitent), Rspamd analitza el **contingut i el comportament** del missatge amb centenars de regles combinades: paraules i patrons típics de spam, si la IP d'origen apareix a llistes negres públiques, si el correu passa o falla SPF/DKIM/DMARC (de fet, també sap comprovar-los ell mateix), l'estructura tècnica del missatge, aprenentatge automàtic (Bayesià) basat en correu que prèviament s'ha marcat com a spam/no-spam, i molt més.

Amb tot això, li assigna una **puntuació numèrica** a cada correu (per exemple, `-2.5` = molt segur; 15 = spam evident) i, segons uns llindars configurables, decideix una acció: deixar-lo passar, afegir-hi una capçalera d'avís, posar-lo en quarantena, o rebutjar-lo directament.

Per què Rspamd i no un altre: és l'estàndard actual en servidors moderns perquè és molt ràpid (escrit en C, mentre que l'alternativa clàssica, SpamAssassin, és en Perl i molt més lenta), i porta integrats de sèrie molts mòduls que altrament necessitarien programes separats (verificació SPF/DKIM/DMARC, llistes grises, detecció Bayesiana…).

Redis

Redis és una base de dades molt ràpida que **guarda dades en memòria** (no és una base de dades relacional com MySQL; és més aviat un magatzem de tipus clau-valor).

Rspamd el fa servir com a **memòria persistent** per a tres coses concretes:

1. **Filtre Bayesià**: quan un usuari marca correu com a spam o no-spam, Rspamd va “aprenent” patrons. Aquest aprenentatge s'ha de desar en algun lloc que sobrevisqui als reinicis del dimoni --- aquí és on entra Redis.
2. **Llistes grises** (*greylisting*): una tècnica antispam que rebutja temporalment el primer intent d'un remitent desconegut (els servidors de spam solen no reintentar-ho, els legítims sí). Redis guarda quins remitents ja s'han vist abans.

3. **Límits de freqüència** (*rate limiting*): per exemple, “aquest usuari no pot enviar més de X correus per minut” --- cal recordar quants n'ha enviat, i això també es desa a Redis.

Sense Redis, Rspamd encara funcionaria (moltes regles no en depenen), però **perdria tota la memòria d'aprenentatge i d'estat** cada cop que es reiniciés el servei --- cosa que faria l'antispam molt menys efectiu amb el temps.

En resum: Rspamd és el “cervell” que analitza i decideix, i Redis és la “memòria” on desa el que va aprenent per prendre decisions cada cop millors.

IMPORTANT

Instal·lar el paquet `redis-server` **no és suficient** perquè Rspamd el faci servir. Cal indicar-li explícitament a Rspamd on trobar-lo (pas 5 d'aquest document); si no, Rspamd funciona igualment (la majoria de regles no depenen de Redis), però el filtre Bayesià i les llistes grises queden inactius silenciosament, sense cap error visible a `systemctl status`.

Requisit previ: cal tenir Postfix ja configurat

Rspamd **no és un servidor de correu**, sinó un motor d'anàlisi que s'enganxa a un servidor de correu ja existent mitjançant el protocol **milter**. No funciona de manera autònoma: necessita que un MTA (en aquest cas, Postfix) li passi cada correu entrant abans d'acceptar-lo, i que sàpiga interpretar la resposta (acceptar, rebutjar, afegir capçalera…).

Per tant, abans de seguir aquest document cal tenir Postfix ja instal·lat i funcionant --- típicament amb, com a mínim:

- Un domini de correu configurat.
- Entrega interna cap a Dovecot ja operativa.
- El servei acceptant correu real (entrant o sortint) sense errors.

Si encara no ho tens muntat, segueix primer: Servidor Mail o bé, Implementació de la màquina correu, abans de continuar amb Rspamd --- els passos següents (`smtpd_milters`, `non_smtpd_milters`) simplement no tenen efecte sense un Postfix real on enganxar-se.

1. Instal·la Rspamd i Redis

```
sudo apt install rspamd redis-server
```

2. Configura el worker que parla amb Postfix (mode milter)

```
sudo nano /etc/rspamd/override.d/worker-proxy.inc
```

```
bind_socket = "/run/rspamd/milter.sock mode=0666";
milter = yes;
timeout = 120s;

upstream "local" {
  default = yes;
  self_scan = yes;
}
```

3. Configura les capçaleres del milter

Per defecte, Rspamd només afegeix capçaleres (`X-Spam-Status`, `X-Spamd-Result`) quan pren alguna acció sobre el correu (avís, quarantena, rebuig). Si la puntuació és baixa i l'acció és “cap acció”, **no hi afegeix res**, cosa que dificulta confirmar visualment que Rspamd ha intervingut. Per forçar-ho sempre, activa-ho al mòdul `milter_headers`:

```
sudo nano /etc/rspamd/local.d/milter_headers.conf
```

```
extended_spam_headers = true;
skip_authenticated = false;
skip_local = false;
```

```
sudo rspamadm configtest
sudo systemctl restart rspamd
```

IMPORTANT

En producció real, val la pena revertir `skip_authenticated`/`skip_local` a `true` (o eliminar-les) per no exposar detalls d'anàlisi interna en correu de sortida legítim; aquí es desactiven només amb finalitat de prova.

4. Restringeix el worker normal a `localhost`

Per defecte escolta a totes les interfícies; no cal exposar-lo fora de la màquina:

```
sudo nano /etc/rspamd/override.d/worker-normal.inc
```

```
bind_socket = "127.0.0.1:11333";
```

5. Defineix les xarxes locals del laboratori

Per no penalitzar el correu intern com si vingués de fora:

```
sudo nano /etc/rspamd/local.d/options.inc
```

```
local_networks = [
  "127.0.0.0/8",
  "10.0.2.0/24"
];
```

6. Connecta Rspamd a Redis

Sense aquest pas, Redis queda instal·lat, però Rspamd no el fa servir:

```
sudo nano /etc/rspamd/local.d/redis.conf
```

```
servers = "127.0.0.1:6379";
```

Activa també l'estadística Bayesiana perquè faci servir Redis com a magatzem:

```
sudo nano /etc/rspamd/local.d/classifier-bayes.conf
```

```
backend = "redis";
servers = "127.0.0.1:6379";
```

NOTA

El filtre Bayesià necessita un mínim de **200 correus d'entrenament** (marcats com a spam/no-spam) abans de començar a influir realment en la puntuació. És normal veure `bayes_classify: not classified... Currently: 0; minimum 200 required` als logs durant els primers dies --- no és cap error.

7. Verifica i arrenca Rspamd

Verifica la configuració:

```
sudo rspamadm configtest
```

Sortida esperada:

```
syntax OK
```

Activa'l perquè s'iniciï automàticament en cada arrencada del sistema i l'arrenqui immediatament:

```
sudo systemctl enable --now rspamd
```

Verifica l'estat del servei:

```
sudo systemctl status rspamd
```

Sortida esperada:

```
● rspamd.service - rapid spam filtering system
     Loaded: loaded (/usr/lib/systemd/system/rspamd.service; enabled;
 ↪  preset: enabled)
     Active: active (running) since Sun 2026-07-19 09:31:05 UTC; 6min
 ↪  ago
 Invocation: b8cc2865745d417388d632dec2dd1f6d
       Docs: https://rspamd.com/doc/
   Main PID: 5213 (rspamd)
      Tasks: 5 (limit: 1718)
     Memory: 59.7M (peak: 59.9M)
        CPU: 531ms
     CGroup: /system.slice/rspamd.service
             ├─5213 "rspamd: main process; 0.1 msg/sec, 0.0 msg/sec
 ↪  spam, 0.1 msg/sec ham; 0.19s avg processing time"
             ├─5215 "rspamd: rspamd_proxy process
 ↪  (/run/rspamd/milter.sock mode=0666)"
             ├─5216 "rspamd: controller process (localhost:11334)"
             ├─5217 "rspamd: normal process (127.0.0.1:11333)"
             └─5218 "rspamd: hs_helper process"

de jul. 19 09:31:05 correu systemd[1]: Started rspamd.service - rapid
 ↪  spam filtering system.
de jul. 19 09:31:06 correu rspamd[5213]: 2026-07-19 09:31:06
 ↪  #5213(main) <da4351>; main; main: rspamd 3.8.1 is loading
 ↪  configuration, build id: release
```

Confirma també que Redis respon i que Rspamd hi ha pogut pujar els seus scripts:

```
redis-cli ping
```

Resposta esperada:

```
PONG
```

8. Verifica que el socket respon

Comprova, sense enviar cap correu de prova, si el socket de Rspamd existeix i està escoltant peticions:

```
nc -zU /run/rspamd/milter.sock && echo "Rspamd milter actiu"
```

Resposta esperada:

```
Rspamd milter actiu
```

9. Comprova el chroot de Postfix

Molts serveis de Postfix (`smtps`, `submission`) corren per defecte dins d'un **chroot** a `/var/spool/postfix/`. Si és el cas, el procés **no veu el sistema de fitxers real** --- quan li dius `unix:/run/rspamd/milter.sock`, en realitat busca `/var/spool/postfix/run/rspamd/milter.sock`, que no existeix, i falla amb `No such file or directory` **encara que el socket real estigui perfectament actiu**.

Comprova-ho:

```
sudo postconf -M | grep -E "smtps|submission"
```

Mira la **cinquena columna** de cada línia (la del chroot):

```
smtps      inet  n       -       y       -       -       smtpd ...
```

Si hi ha una **y**, cal desactivar-lo per a cada servei afectat:

```
sudo nano /etc/postfix/master.cf
```

Canvia la y per n a la cinquena columna de les línies `smtps` i `submission`:

```
smtps      inet  n       -       n       -       -       smtpd ...
submission inet  n       -       n       -       -       smtpd ...
```

```
sudo systemctl restart postfix
```

NOTA

Si en reiniciar Postfix torna a sortir l'avís `warning: not owned by root: /var/spool/postfix/etc/resolv.conf`, no té relació amb el milter --- és un problema independent de permisos dins del chroot. Corregeix-lo amb:

```
sudo cp /etc/resolv.conf /var/spool/postfix/etc/resolv.conf
sudo chown root:root /var/spool/postfix/etc/resolv.conf
sudo chmod 644 /var/spool/postfix/etc/resolv.conf
```

10. Connecta Rspamd a Postfix com a milter

Configura Postfix perquè abans d'acceptar definitivament un correu entrant, l'enviï a Rspamd perquè l'analitzi:

```
sudo postconf -e 'smtpd_milters = unix:/run/rspamd/milter.sock'
```

Aquesta línia és la parella de l'anterior, però per a un cas diferent: correu que no arriba per xarxa, sinó que es genera localment al mateix servidor:

```
sudo postconf -e 'non_smtpd_milters = unix:/run/rspamd/milter.sock'
```

Defineix què ha de fer Postfix si Rspamd no respon --- és a dir, un pla de contingència per a quan el milter falla o queda inaccessible:

```
sudo postconf -e 'milter_default_action = accept'
```

IMPORTANT

`milter_default_action = accept` és important: si Rspamd no respon (per manteniment, per exemple), Postfix continuarà entregant correu en lloc de bloquejar-lo tot. El compromís: durant una eventual caiguda de Rspamd, el correu flueix sense filtrar en lloc de quedar completament bloquejat.

Reinicia Postfix:

```
sudo systemctl restart postfix
```

AVÍS

Si en el futur afegeixes OpenDKIM: no substitueixis aquesta línia, encadena'ls separats per comes:

```
smtpd_milters = local:/run/opendkim/opendkim.sock,
 ↪  unix:/run/rspamd/milter.sock
```

Postfix crida els milters en l'ordre indicat.

11. Verifica amb un correu de prova

Envia un correu de prova entre dos usuaris als quals tinguis accés.

Obre el missatge rebut. Hauries de veure capçaleres com aquestes:

X-Rspamd-Queue-Id: AB16DA06AA

X-Rspamd-Server: correu

X-Spamd-Result: default: False [0.46 / 15.00];

```
R_MIXED_CHARSET(0.56)[];
MIME_GOOD(-0.10)[multipart/alternative,text/plain];
TO_DOM_EQ_FROM_DOM(0.00)[];
MID_RHS_MATCH_FROM(0.00)[];
RCVD_COUNT_ZERO(0.00)[0];
ARC_NA(0.00)[];
RCPT_COUNT_ONE(0.00)[1];
FROM_HAS_DN(0.00)[];
TO_DN_ALL(0.00)[];
URIBL_BLOCKED(0.00)[correu.thos.local:rdns,correu.thos.loc
 ↪  al:helo];
FROM_EQ_ENVFROM(0.00)[];
TO_MATCH_ENVRCPT_ALL(0.00)[];
MIME_TRACE(0.00)[0:+,1:+,2:~]
```

X-Rspamd-Action: no action

Original del missatge sencer:

```
Return-Path: <joan@thos.local>
Delivered-To: maria@thos.local
Received: from correu.thos.local
    by correu.thos.local with LMTP
    id OIPLNI6fXGqAFQAAfUf1FQ
    (envelope-from <joan@thos.local>)
    for <maria@thos.local>; Sun, 19 Jul 2026 09:57:34 +0000
Received: from correu.thos.local (correu.thos.local [10.0.2.10])
    (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits)
     key-exchange x25519 server-signature RSA-PSS (2048 bits)
 ↪  server-digest SHA256)
    (No client certificate requested)
    by correu.thos.local (Postfix) with ESMTPSA id AB16DA06AA
    for <maria@thos.local>; Sun, 19 Jul 2026 09:57:34 +0000 (UTC)
From: "Joan Garcia" <joan@thos.local>
To: "Maria Puig" <maria@thos.local>
User-Agent: SOGoMail 5.12.4
MIME-Version: 1.0
Date: Sun, 19 Jul 2026 11:57:34 +0200
Subject: Missatge de prova
```

```
Message-ID: <4f4a40e4-6d6f-0303-7098-cb6f3e38e4e1@thos.local>
X-Forward: 10.0.2.14
Content-Type: multipart/alternative; boundary="----=_=-_OpenGroupwar
 ↪  e_org_NGMime-1879-1784455054.509937-1------"
X-Rspamd-Queue-Id: AB16DA06AA
X-Rspamd-Server: correu
X-Spamd-Result: default: False [0.46 / 15.00];
    R_MIXED_CHARSET(0.56)[];
    MIME_GOOD(-0.10)[multipart/alternative,text/plain];
    TO_DOM_EQ_FROM_DOM(0.00)[];
    MID_RHS_MATCH_FROM(0.00)[];
    RCVD_COUNT_ZERO(0.00)[0];
    ARC_NA(0.00)[];
    RCPT_COUNT_ONE(0.00)[1];
    FROM_HAS_DN(0.00)[];
    TO_DN_ALL(0.00)[];
    URIBL_BLOCKED(0.00)[correu.thos.local:rdns,correu.thos.local:hel
 ↪  o];
    FROM_EQ_ENVFROM(0.00)[];
    TO_MATCH_ENVRCPT_ALL(0.00)[];
    MIME_TRACE(0.00)[0:+,1:+,2:~]
X-Rspamd-Action: no action

------=_=-_OpenGroupware_org_NGMime-1879-1784455054.509937-1------
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 53


Bon dia, Maria.
Aquest és un missatge de prova.

------=_=-_OpenGroupware_org_NGMime-1879-1784455054.509937-1------
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 81

<html><p>Bon dia, Maria.</p><p>Aquest és un missatge de prova.</p>=
</html>

------=_=-_OpenGroupware_org_NGMime-1879-1784455054.509937-1--------
```

12. Prova de bloqueig real amb el patró GTUBE

Rspamd reconeix un patró de test estàndard anomenat **GTUBE** sense necessitat d'enviar spam de veritat.

La **GTUBE** (*Generic Test for Unsolicited Bulk Email*) és una cadena de text estàndard, dissenyada específicament perquè qualsevol filtre antispam la reconegui i la marqui **sempre** com a spam, sense ambigüitat.

12.1. Per què existeix GTUBE?

Quan proves un filtre antispam, tens un problema pràctic: **no vols enviar spam real** per comprovar que el filtre funciona (seria irresponsable, i a més podria acabar afectant la reputació del teu propi servidor). La GTUBE resol això: és un patró **universalment reconegut** pels motors antispam (SpamAssassin la va crear originàriament, però Rspamd i pràcticament tots els altres també la reconeixen) com “això és una prova, tracta-ho com a spam garantit”.

12.2. Com funciona?

No és cap tècnica sofisticada --- és literalment una **regla explícita** dins del motor: “si el cos del missatge conté exactament aquesta cadena, assigna-li la puntuació màxima de spam”. No hi ha cap anàlisi real de contingut, patrons, o llistes negres involucrades; és una porta del darrere deliberada i documentada per a fer proves.

12.3. Com s'ha de fer servir?

Cal enganxar-la **exactament tal com és**, sense cap salt de línia enmig ni espais addicionals, en algun lloc del **cos del missatge** (no funciona a l'assumpte):

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

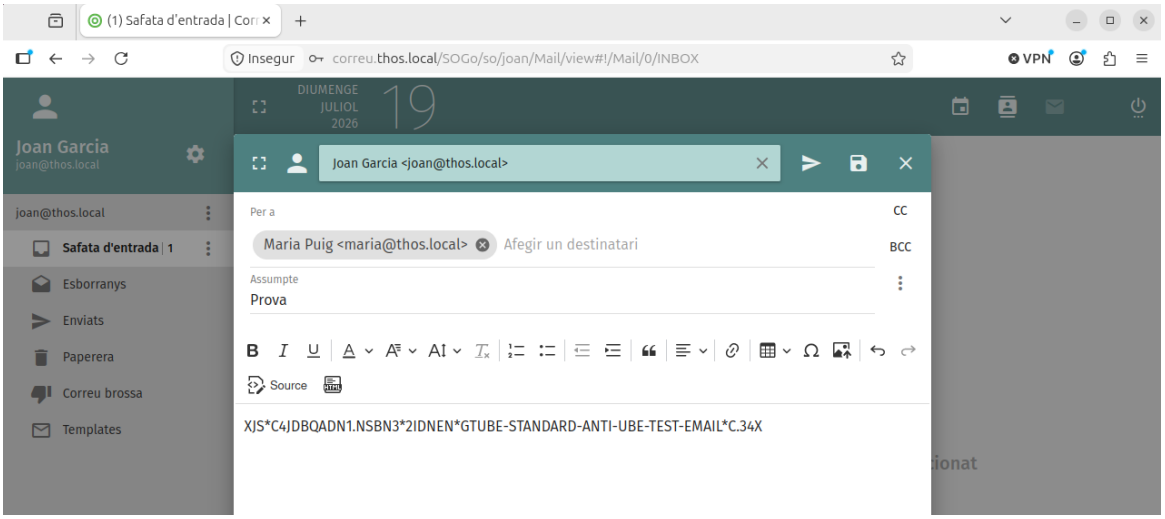

Figura 2: Missatge amb el patró GTUBE

12.4. Què hauries de veure si Rspamd funciona correctament

Als logs (`rspamd.log`), una puntuació molt per sobre del llindar de rebuig.

```
sudo tail -n 20 /var/log/rspamd/rspamd.log
```

En aquest cas la puntuació és 15.00

```
2026-07-19 10:24:03 #5479(rspamd_proxy) <563f09>; proxy;
 ↪  rspamd_add_passthrough_result:
 ↪  <48d4e64b-9e56-9a0a-5ee6-918d711e9641@thos.local>: set pre-result
 ↪  to 'reject' (15.00): 'Gtube pattern' from GTUBE(3)
2026-07-19 10:24:03 #5479(rspamd_proxy) <563f09>; proxy;
 ↪  rspamd_task_write_log: id:
 ↪  <48d4e64b-9e56-9a0a-5ee6-918d711e9641@thos.local>, qid:
 ↪  <0898EA06BD>, ip: 10.0.2.10, user: joan, from: <joan@thos.local>,
 ↪  (default: S (reject): [15.00/15.00] [GTUBE(0.00){}]), len: 927,
 ↪  time: 0.812ms, dns req: 0, digest:
 ↪  <24446c0f37d3f221f11c8e75037a1273>, rcpts: <maria@thos.local>,
 ↪  mime_rcpts: <maria@thos.local>, forced: reject "Gtube pattern";
 ↪  score=15.00 (set by GTUBE)
```

Segons com tinguis configurada l'acció per defecte de Rspamd, el correu es rebutjarà directament (l'emissor rebrà un error SMTP) o arribarà marcat clarament com a spam

(per exemple, a la carpeta de Correu brossa de SOGo si tens `sieve` filtrant per `X-Spam-Status`). En aquest cas, ni tan sols s'envia:

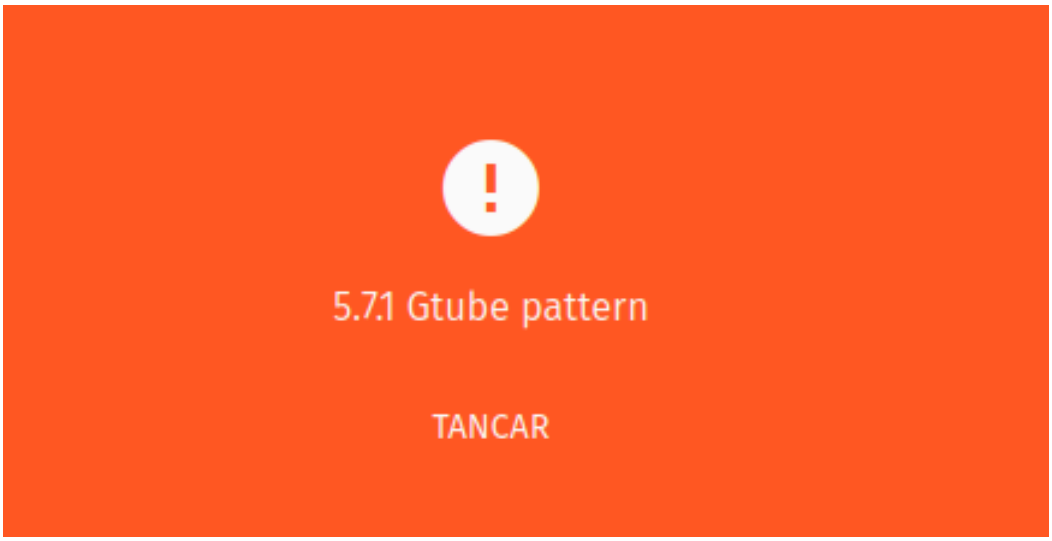

Figura 3: Patró GTUBE detectat

13. Interfície web de monitoratge (Opcional)

Mostra en temps real cada correu analitzat i la puntuació desglossada per regla:

```
rspamadm pw
```

Copia la contrasenya xifrada resultant:

```
sudo nano /etc/rspamd/override.d/worker-controller.inc
```

```
password = "$2$...";
bind_socket = "127.0.0.1:11334";
```

Reinicia Rspamd:

```
sudo systemctl restart rspamd
```

Accessible a `http://127.0.0.1:11334` --- fes-hi un túnel SSH o un proxy des d'Apache si vols accedir des del navegador.

Per exemple, des del client amb interfície gràfica crea un túnel SSH:

```
ssh -L 11334:127.0.0.1:11334 usuari@correu.thos.local
```

Amb el túnel actiu, obre al navegador del client:

```
http://localhost:11334
```

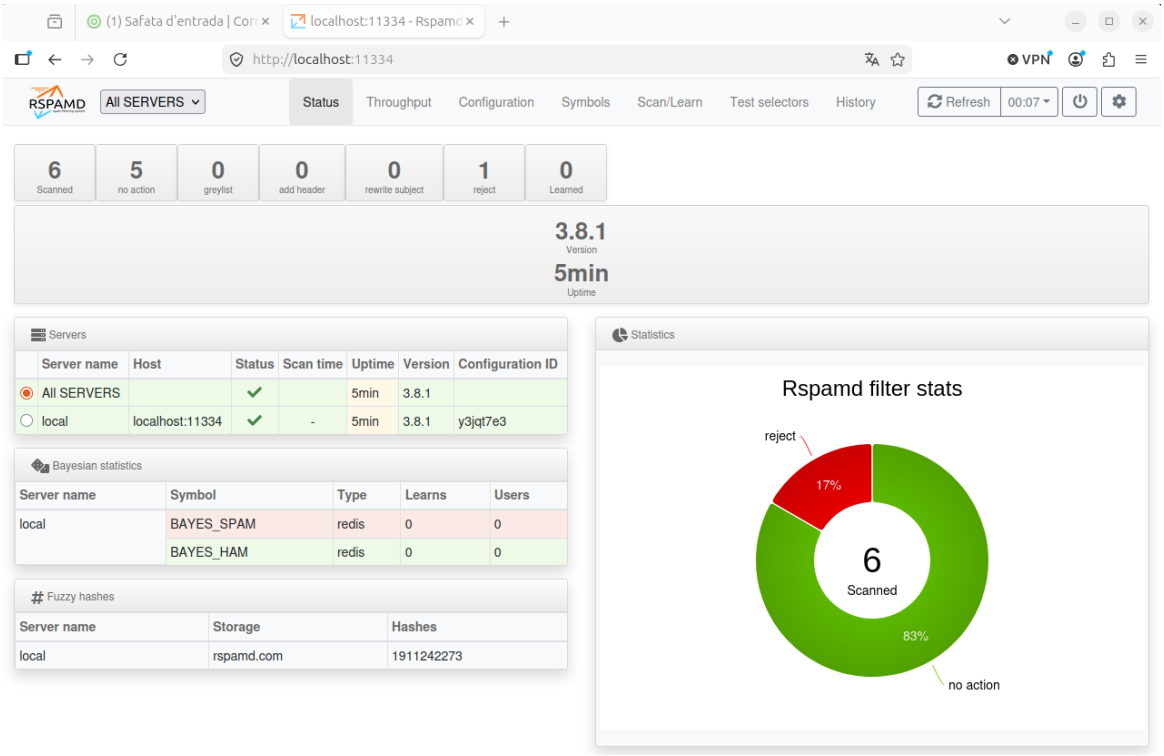

Figura 4: Interfície de monitoratge

Índex de figures

Versions d'aquest document

- HTML
- PDF
- ODT
- MD