MailCow: częste problemy

Poziom: operacyjny

Dotyczy: MailCow Dockerized 2026-07a na docker01, Rocky Linux 9 (sprawdzono w dokumentacji i kodzie wydania 3.10.2026)

Czas: ~30 min

Zestawienie objawów, z którymi zespół IT trafia do MailCow najczęściej: poczta w spamie u odbiorcy, zapełniony dysk, kontener, który przestał być zdrowy, certyfikat, wolny webmail i błędy po aktualizacji. Tabela daje szybką odpowiedź, a trzy procedury pod nią prowadzą krok po kroku przez najczęstsze przypadki.

Skróty użyte na tej stronie (20)

Skrót	Co znaczy
ACME	Automatic Certificate Management Environment: protokół, którym Traefik albo certbot automatycznie zamawiają i odnawiają certyfikat w Let's Encrypt, bez ręcznego wniosku
CAA	Certification Authority Authorization: rekord DNS, który mówi, które urzędy certyfikacji mogą wystawiać certyfikaty dla domeny
ClamAV	Clam AntiVirus: darmowy skaner wirusów, który w MailCow sprawdza załączniki wiadomości i zajmuje przy tym około 1,3 GB pamięci
DKIM	DomainKeys Identified Mail: cyfrowy podpis wiadomości e-mail
DMARC	Domain-based Message Authentication, Reporting and Conformance: reguła domeny mówiąca, co robić z mailem, który nie przeszedł SPF i DKIM
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GiB	gibibajt: 1024 MiB, czyli około 1,07 GB; tej jednostki używa dokumentacja MailCow przy wymaganiach pamięci
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
IMAP	Internet Message Access Protocol: protokół, którym klient pocztowy pobiera wiadomości z serwera, zostawiając je tam zamiast kasować
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
PTR	rekord DNS odwrotny: zamienia adres IP na nazwę
RAM	Random Access Memory: pamięć operacyjna

Skrót	Co znaczy
RBL	Realtime Blackhole List: lista adresów IP, z których szła niechciana poczta; wiele serwerów odrzuca pocztę z adresów znajdujących się na takiej liście
SMTP	Simple Mail Transfer Protocol: protokół wysyłania poczty
SOGo	webmail z kalendarzem i kontaktami, który w MailCow jest dostępny w przeglądarce obok panelu administracyjnego
SPF	Sender Policy Framework: rekord DNS mówiący, które serwery mogą wysyłać pocztę domeny
SSL	Secure Sockets Layer: dawna nazwa szyfrowania, dziś formalnie TLS
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
TTL	Time To Live: czas życia rekordu lub danych

Fiszka

Komendy wykonujemy na `docker01` z konta z uprawnieniami sudo, w katalogu `/docker/apps/mailcow`.

□□ TYLKO ODCZYT

Sprawdzenie stanu kontenerów: docker compose ps

```
cd /docker/apps/mailcow
sudo docker compose ps
```

Pierwszy krok niemal każdej diagnozy. Szukamy kontenerów, które nie mają w kolumnie stanu `healthy` albo restartują się w kółko. Szersze omówienie jest na stronie MailCow: co to jest i z czego się składa.

□□ **cd** = change directory: przejdź do katalogu · **docker** = narzędzie do kontenerów

□□ TYLKO ODCZYT

Sprawdzenie logów jednego kontenera: docker compose logs

```
cd /docker/apps/mailcow
sudo docker compose logs --tail 100 NazwaKontenera
```

(gdzie `NazwaKontenera` to nazwa usługi z pliku Compose, na przykład `nginx-mailcow`, `dovecot-mailcow` albo `acme-mailcow`)

Ostatnie sto linii zwykle wystarcza, żeby zobaczyć przyczynę. Dodanie `-f` śledzi log na żywo.

cd = change directory: przejdź do katalogu · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Sprawdzenie zużycia pamięci przez kontenery: docker stats

```
sudo docker stats --no-stream
```

Jedna migawka zużycia procesora i pamięci każdego kontenera. Interesuje nas kolumna zużycia pamięci wobec limitu, zwłaszcza dla `clamd-mailcow`, `mysql-mailcow` i `sogo-mailcow`.

docker = narzędzie do kontenerów

TYLKO ODCZYT

Sprawdzenie pamięci hosta: free -h

```
free -h
```

Dokumentacja MailCow podaje minimum 6 GiB RAM i 1 GiB swapu, a `docker01` dzieli maszynę z innymi aplikacjami, więc patrzymy na wartość w kolumnie `available`, nie na `free`.

GiB = gibibajt: 1024 MiB, czyli około 1,07 GB; tej jednostki używa dokumentacja MailCow przy wymaganiach pamięci · **RAM** = Random Access Memory: pamięć operacyjna · **free** = pokaż ile pamięci RAM i przestrzeni wymiany jest wolne, a ile zajęte

TYLKO ODCZYT

Sprawdzenie wolnego miejsca na /docker: df -h

```
df -h /docker
```

Przykład:

```
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vgDocker-lvDocker  200G  189G   11G  95% /docker
```

df = disk free: ile miejsca wolnego na dyskach

TYLKO ODCZYT

Sprawdzenie, co zajmuje miejsce pod /docker: du

```
sudo du -xh --max-depth=1 /docker | sort -h
```

Przełącznik `-x` ogranicza liczenie do tego jednego wolumenu. Skrzynki, baza i klucze MailCow leżą w wolumenach Dockera pod `/docker/lib/volumes`, a lokalne kopie w `/docker/kopie/mailcow`. Opis różnic między `df` a `du` jest na stronie o dyskach w książce o Linuksie.

du = disk usage: ile miejsca zajmuje katalog

TYLKO ODCZYT

Sprawdzenie dat ważności certyfikatów: expiry-dates.sh

```
cd /docker/apps/mailcow
bash helper-scripts/expiry-dates.sh
```

Skrypt z repozytorium MailCow pyta o certyfikat po SMTP, IMAP i HTTPS i wypisuje datę wygaśnięcia każdego z nich.

Przykład (daty przykładowe):

```
TLS expiry dates:
Postfix: Dec 11 08:12:45 2026 GMT
Dovecot: Dec 11 08:12:45 2026 GMT
Nginx:   Dec 11 08:12:45 2026 GMT
```

SMTP = Simple Mail Transfer Protocol: protokół wysyłania poczty · **IMAP** = Internet Message Access Protocol: protokół, którym klient pocztowy pobiera wiadomości z serwera, zostawiając je tam zamiast kasować · **HTTPS** = HTTP zabezpieczony szyfrowaniem TLS · **cd** = change directory: przejdź do katalogu · **bash** = Bourne Again Shell: uruchom powłokę bash albo wskazany w niej skrypt

TYLKO ODCZYT

Sprawdzenie, czy adres jest na liście RBL: dig

```
dig +short 10.113.0.203.NazwaListyRBL
```

(gdzie `NazwaListyRBL` to nazwa domeny listy, którą chcemy sprawdzić, a `10.113.0.203` to nasz adres `203.0.113.10` zapisany od tyłu)

Odpowiedź w postaci adresu `127.0.0.x` oznacza, że adres jest na liście. Brak odpowiedzi znaczy, że listy nie ma. Dokumentacja MailCow poleca do przeglądu wielu list serwisy zbiorcze, na przykład dnsbl.info.

RBL = Realtime Blackhole List: lista adresów IP, z których szła niechciana poczta; wiele serwerów odrzuca pocztę z adresów znajdujących się na takiej liście · **dig** = domain information groper: odpytywanie DNS

TYLKO ODCZYT

Sprawdzenie, kto wysyła najwięcej poczty: grep sasl_username

```
cd /docker/apps/mailcow
sudo docker compose logs --no-log-prefix --tail 20000 postfix-mailcow | grep -o
'sasl_username=[^,]*' | sort | uniq -c | sort -rn | head
```

Zestawienie zalogowanych kont według liczby połączeń z ostatnich dwudziestu tysięcy linii logu. Konto, które wyraźnie wyprzedza resztę, jest podejrzane o przejęcie.

cd = change directory: przejdź do katalogu · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Restart jednego kontenera: docker compose restart

```
cd /docker/apps/mailcow
sudo docker compose restart NazwaKontenera
```

Restart pojedynczego kontenera nie rusza pozostałych. Do restartu SOGo mamy też pozycję `E-Mail → Restart SOGo` w panelu.

SOGo = webmail z kalendarzem i kontaktami, który w MailCow jest dostępny w przeglądarce obok panelu administracyjnego · **cd** = change directory: przejdź do katalogu · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Wymuszenie odnowienia certyfikatu: force_renew

```
cd /docker/apps/mailcow
sudo touch data/assets/ssl/force_renew
sudo docker compose restart acme-mailcow
```

Plik `force_renew` każe kontenerowi ACME spróbować odnowienia od razu, zamiast czekać na codzienne sprawdzenie. Po restarcie czytamy log `acme-mailcow`, żeby zobaczyć wynik.

ACME = Automatic Certificate Management Environment: protokół, którym Traefik albo certbot automatycznie zamawiają i odnawiają certyfikat w Let's Encrypt, bez ręcznego wniosku · **cd** = change directory: przejdź do katalogu · **touch** = utwórz pusty plik albo uaktualnij jego datę modyfikacji, jeśli już istnieje · **docker** = narzędzie do kontenerów

Z życia wzięte

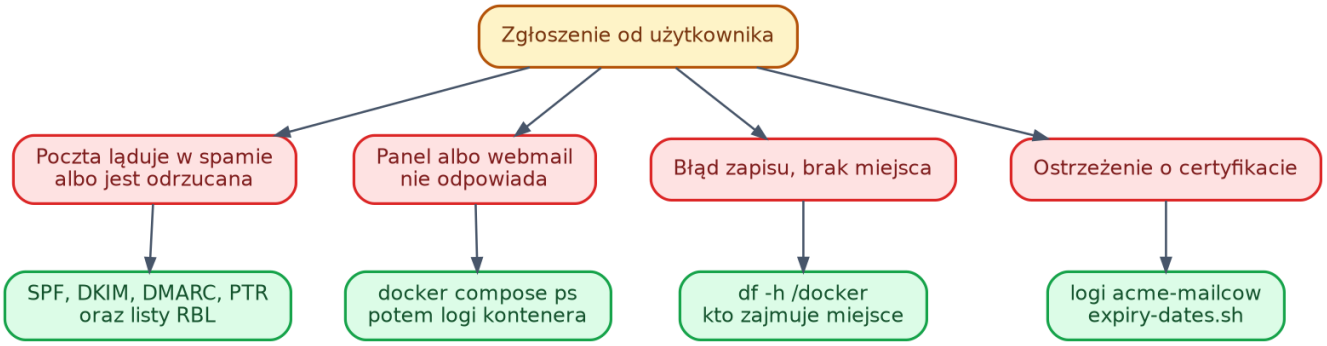
Z ŻYCIA WZIĘTE

Środa, 6:05. Marek ze Swarzędza dzwoni, że webmail nie otwiera się, a skaner z wysyłką dokumentów zgłasza błąd poczty. Tomek robi `df -h /docker` i widzi 100 procent: po sobotniej aktualizacji zostały stare obrazy, a trzy nocne kopie lokalne zajęły resztę wolnego miejsca. Baza MailCow przewróciła się przy próbie zapisu, więc `mysql-mailcow` nie jest zdrowy, a reszta kontenerów czeka na niego. Tomek zwalnia miejsce według procedury 3 poniżej i przed 7:00 poczta działa.

Jak to działa

MailCow składa się z kilkunastu kontenerów, które zależą od siebie jak ogniwa łańcucha: Postfix czeka na bazę i Rspamd, Dovecot na bazę, SOGo na bazę i Dovecota. Dlatego jedna awaria potrafi wyglądać jak awaria całego systemu, a przyczynę znajdujemy zwykle w pierwszym kontenerze od dołu łańcucha, nie w tym, który użytkownik zgłasza.

Pojęcie: Watchdog. Kontener `watchdog-mailcow`, który co jakiś czas sprawdza pozostałe usługi (Postfix, Dovecot, Nginx, MySQL, Rspamd, SOGo, ClamAV i inne). Gdy usługa nie odpowiada kilka razy z rzędu (próg zależy od usługi, domyślnie od kilku do kilkunastu prób), Watchdog sam restartuje kontener. Powiadomienia mailem włączamy w `mailcow.conf` zmienną `WATCHDOG_NOTIFY_EMAIL`, a historię jego decyzji czytamy w `System → Information → Logs → Watchdog`.



Cztery typowe zgłoszenia i pierwsze miejsce, w które zaglądamy przy każdym z nich (kliknięcie otwiera pełny rozmiar)

Pierwsze pytanie brzmi zawsze: czy poczta jest wciąż dostarczana, czy tylko przestał działać panel albo webmail. Awaria `nginx-mailcow` odcina panel, ale Postfix i Dovecot pracują dalej, o czym przekonuje scenka na stronie MailCow: co to jest i z czego się składa. Awaria bazy zatrzymuje wszystko.

Częste błędy

Objaw	Przyczyna	Co robimy
Nasza poczta ląduje w spamie u odbiorcy	niezgodność SPF, DKIM, DMARC albo brak PTR dla `203.0.113.10`	procedura 1 niżej; PTR ustawia dostawca łącza, nie my
Odbiorca odrzuca nas komunikatem o liście blokad (RBL)	nasz adres jest na liście, zwykle przez przejęte konto albo wysyłkę masową	sprawdzamy listy (`dig` wyżej), szukamy przejętego konta, dopiero potem prosimy o usunięcie z listy
`/docker` pełny, kontenery niezdrowe	trzy lokalne kopie, stare obrazy po aktualizacji, rosnące skrzynki	procedura 3 niżej
Kontener `unhealthy` albo `Restarting`	zajęty port, brak plików SSL, brak miejsca, zła wersja obrazu, zapora między kontenerami	procedura 2 niżej
Certyfikat nie odnawia się, przeglądarka ostrzega	port 80 niedostępny z internetu (zapora `fw01`), brak rekordów `autodiscover` i `autoconfig` albo rekord CAA blokujący Let's Encrypt, własny certyfikat w `data/assets/ssl`	`docker compose logs --tail 200 acme-mailcow`, sprawdzenie reguły przekierowania portu 80 na `fw01` i rekordów DNS, potem `force_renew`
Webmail SOGo działa wolno albo się zawiesza	za mało pamięci hosta, dużo urządzeń z synchronizacją ActiveSync, przeciążona baza	`docker stats` i `free -h`, log SOGo w `System → Information → Logs → SOGo`, restart SOGo z menu, a przy braku pamięci jej dołożenie

Objaw	Przyczyna	Co robimy
Host zaczyna używać swapu, a najwięcej pamięci zajmuje `clamd-mailcow`	ClamAV wczytuje całą bazę sygnatur do pamięci (według dokumentacji około 1,3 GB przy rozszerzonych bazach)	dokładamy pamięć albo wyłączamy skanowanie (`SKIP_CLAMD=y` w `mailcow.conf`, decyzja opisana niżej)
Po aktualizacji błędy `Invalid interpolation format` albo `NoneType`	zbyt stary Docker albo wtyczka Compose	wymagane Docker 24 lub nowszy i Compose 2 lub nowszy, aktualizujemy je
Po aktualizacji kontenery `unhealthy` albo panel pokazuje błąd	niedokończona aktualizacja, niezgodne wersje kodu i obrazów	`docker compose pull`, potem `docker compose up -d`, logi niezdrowego kontenera
`update.sh` kończy się przy sprawdzaniu łączności	zapora blokuje ICMP z serwera	`update.sh --skip-ping-check`
Dovecot restartuje się w pętli	brakuje plików `cert.pem`, `key.pem` albo `dhparams.pem` w `data/assets/ssl`	przywracamy je według strony o resecie certyfikatów w dokumentacji MailCow albo z kopii katalogu instalacji z `bkp01`
Rspamd nie startuje po przeniesieniu na inną maszynę	pamięć podręczna reguł (pliki `*.hs` i `*.hsmp` w `/var/lib/rspamd`) pochodzi z innej architektury procesora	usuwamy te pliki w kontenerze i restartujemy `rspamd-mailcow`

Co do ClamAV: wyłączenie skanowania oszczędza około 1,3 GB, ale oznacza brak skanowania wirusów po stronie serwera pocztowego. Czy to akceptowalne, zależy od tego, czy stacje mają własną ochronę, i jest decyzją autora, nie czymś, co zmieniamy przy okazji awarii. Wyłączenie wymaga ponownego złożenia kontenerów (`docker compose up -d`).

Procedura 1: poczta ląduje w spamie u odbiorcy

Najczęstszy problem z pocztą. Serwer odbiorcy ocenia nas po tym, co widzi w DNS, więc zaczynamy od DNS, nie od MailCow.

1. Prosimy odbiorcę o pełne nagłówki wiadomości i szukamy w nich linii `Authentication-Results`, w której stoją wyniki `spf=`, `dkim=` i `dmarc=`. Każde `fail` albo `none` wskazuje, który mechanizm zawodzi.
2. Sprawdzamy rekordy z zewnątrz sieci Zielonej:

```
dig +short TXT zielona.example
dig +short TXT dkim._domainkey.zielona.example
dig +short TXT _dmarc.zielona.example
dig +short -x 203.0.113.10
```

1. Otwieramy `E-Mail → Configuration → Domains`, w wierszu domeny klikamy przycisk DNS i porównujemy kolumny „Correct Data" i „Current State".
2. Naprawiamy według objawu. Brak adresu `203.0.113.10` w SPF poprawiamy w DNS. Brak albo zły klucz DKIM publikujemy ponownie z `System → Configuration → Options → ARC/DKIM keys`. Brak PTR albo PTR niezgodny z nazwą `poczta.zielona.example` zgłaszamy dostawcy łącza, bo nie da się go ustawić w naszej strefie. Gdy kłopot ma tylko poczta z innego systemu (Optima, skaner, sklep, newsletter), sprawdzamy go według tabeli nadawców na stronie SPF, DKIM i DMARC: trzy podejścia do poczty i systemy wysyłające w imieniu domeny, a źródła, które nie przechodzą DMARC, pokazują raporty opisane na stronie Droga DMARC od p=none do p=reject z raportami parsedmarc na docker01.
3. Sprawdzamy listy RBL (komenda `dig` w Fiszce). Jeżeli adres jest na liście, szukamy przyczyny, na przykład konta wysyłającego niepokojąco dużo poczty.
4. Po poprawce wysyłamy wiadomość testową na zewnętrzny adres i sprawdzamy w nagłówkach `spf=pass`, `dkim=pass`, `dmarc=pass`. Po zmianach w DNS czekamy co najmniej do wygaśnięcia TTL.

Procedura 2: kontener jest niezdrowy

Zasada: zaczynamy od logów jednego kontenera, nie od restartu całego stosu.

1. `sudo docker compose ps` w katalogu `/docker/apps/mailcow`: zapisujemy, które kontenery są niezdrowe albo restartują się.
2. `sudo docker compose logs --tail 100 NazwaKontenera` dla pierwszego w łańcuchu zależności, zwykle `mysql-mailcow`, potem `redis-mailcow`, dopiero potem reszty.
3. Dopasowujemy log do przyczyny:

- `nginx-mailcow` restartuje się, a w logu jest `address already in use`: port 80 albo 443 zajmuje inny proces, sprawdzamy `sudo ss -tlnp | grep -E ":80|:443"`.
- `dovecot-mailcow` restartuje się, a w logu brakuje plików SSL: przywracamy `cert.pem`, `key.pem`, `dhparams.pem` w `data/assets/ssl`.
- `mysql-mailcow` nie startuje: najpierw sprawdzamy miejsce na dysku (procedura 3).
- kilka kontenerów naraz ma kłopot z łącznością: sprawdzamy, czy zapora hosta nie blokuje sieci kontenerów MailCow (domyślnie `172.22.1.0/24`).

1. Po naprawie przyczyny restartujemy tylko ten kontener: `sudo docker compose restart NazwaKontenera`. Cały stos zatrzymujemy dopiero wtedy, gdy zależności tego wymagają, bo zrywa to sesje wszystkich użytkowników.
2. Sprawdzamy, że `docker compose ps` pokazuje wszystkie kontenery jako zdrowe, a Watchdog w `System → Information → Logs → Watchdog` przestał zgłaszać błędy.

Procedura 3: dysk /docker jest pełny

1. `df -h /docker` i `sudo du -xh --max-depth=1 /docker | sort -h`: ustalamy, który katalog rośnie. Dla Dockera dodatkowo `sudo docker system df`.
2. Zwalniamy miejsce od rzeczy, które da się odtworzyć. Najpierw stare obrazy po aktualizacji:

```
cd /docker/apps/mailcow
sudo ./update.sh --gc
```

1. Potem najstarsze lokalne kopie MailCow, ale tylko takie, które są już na `bkp01`. Sprawdzamy to, zanim cokolwiek usuniemy:

```
ls -1 /docker/kopie/mailcow
ssh nazwa_usera@bkp01 "ls -1 /backup/mailcow"
```

1. Usuwamy wybrany katalog kopii poleceniem `sudo rm -rf /docker/kopie/mailcow/NazwaKatalogu` i sprawdzamy `df -h /docker`.
2. Nigdy nie usuwamy plików ręcznie w `/docker/lib/volumes`, bo tam leżą skrzynki, baza i klucze DKIM, a ręczne kasowanie niszczy spójność danych.
3. Gdy nic z powyższego nie wystarcza, powiększamy wolumen `vgDocker/lvDocker`, według strony o powiększaniu przestrzeni w książce o Linuksie, i rozważamy krótszy `--delete-days` w skrypcie kopii.
4. Po odzyskaniu miejsca uruchamiamy ponownie niezdrowe kontenery, zaczynając od `mysql-mailcow`, i wykonujemy `docker compose ps`.

□□ Ćwiczenia

1. Użytkownicy zgłaszają, że webmail nie otwiera się, ale pocztę w Outlooku wciąż odbierają. Który kontener podejrzewamy w pierwszej kolejności i dlaczego poczta jednak działa?

Odpowiedź

`nginx-mailcow`, który wystawia panel i webmail na portach 80 i 443. Outlook łączy się z Dovecotem po IMAP i wysyła przez Postfixa, a te kontenery nie przechodzą przez Nginxa, więc pracują niezależnie od niego.

1. Odbiorca z biura rachunkowego odrzuca naszą pocztę, a `dig` pokazuje poprawne SPF, DKIM i DMARC. Co sprawdzamy dalej?

Odpowiedź

Rekord PTR dla `203.0.113.10` oraz to, czy adres nie jest na liście RBL. Poprawny SPF, DKIM i DMARC nie wystarczą, jeśli odwrotny DNS nie zgadza się z nazwą `poczta.zielona.example` albo adres ma złą reputację.

1. Dlaczego przy pełnym dysku zaczynamy od `update.sh --gc` i najstarszych kopii, a nie od kasowania w `/docker/lib/volumes`?

Odpowiedź

Stare obrazy i lokalne kopie da się odtworzyć (obrazy pobraniem, kopie z `bkp01`). W `/docker/lib/volumes` leżą żywe dane poczty, których ręczne usunięcie niszczy skrzynki albo bazę i można je odzyskać tylko z kopii.

1. Certyfikat wygasa za tydzień, a `acme-mailcow` w logu pisze, że nie może zweryfikować żadnej nazwy. Jakie trzy rzeczy sprawdzamy?

Odpowiedź

Czy port 80 jest przekierowany z `203.0.113.10` na `docker01` (zapora `fw01`), czy rekord `A` dla `poczta.zielona.example` oraz `CNAME` dla `autodiscover` i `autoconfig` wskazują na ten adres, i czy rekord CAA domeny nie zabrania Let's Encrypt wystawiania certyfikatów. Dopiero po poprawce wymuszamy odnowienie plikiem `force_renew`.

Powiązane: MailCow: co to jest i z czego się składa · Instalacja MailCow na docker01 od zera · MailCow: codzienna praca z domenami, skrzynkami i spamem · MailCow: kopia zapasowa i aktualizacja · Diagnostyka dostarczania · Utrzymanie hosta dockerowego · SPF, DKIM i DMARC: trzy podejścia do poczty i systemy wysyłające w imieniu domeny · Droga DMARC od p=none do p=reject z raportami parsedmarc na docker01 · MTA-STS i TLS-RPT: szyfrowanie poczty między serwerami, a BIMI informacyjnie

Źródła: docs.mailcow.email: General Troubleshooting (common problems, logs, reset TLS), Prerequisites (system, DNS), SSL/ACME, Watchdog thresholds, ClamAV additional databases · `helper-scripts/expiry-dates.sh` i `data/Dockerfiles/acme/acme.sh` w github.com/mailcow/mailcow-dockerized

Revision #14
Created 2026-10-03 12:08:22 UTC by Tomasz Konieczny
Updated 2026-10-26 14:20:42 UTC by Tomasz Konieczny