

# PostgreSQL Güvenlik Hardening

Production Veritabanları için Uçtan Uca Güvenlik Sıkılaştırma Rehberi

PostgreSQL 15 / 16 / 17 uyumlu

Güncelleme: Ekim 2026 · Production-grade



PG

## 01 Kimlik Doğrulama pg\_hba.conf EN KRİTİK



| METOT         | DURUM    | AÇIKLAMA                                            |
|---------------|----------|-----------------------------------------------------|
| scram-sha-256 | ÖNERİLEN | Challenge-response, tuzlu hash. Varsayılan (PG14+). |
| cert          | ÖNERİLEN | İstemci sertifikası ile doğrulama (hostssl).        |
| peer          | Lokal    | Unix socket — OS kullanıcı adı eşleşmesi.           |
| md5           | ESKİ     | Zayıf hash, replay riski. PG18'de deprecated.       |
| password      | YASAK    | Parola ağda düz metin olarak gider.                 |
| trust         | YASAK    | Parolasız giriş! Sadece izole localhost.            |

### GÜVENLİ PG\_HBA.CONF

```
# TYPE DATABASE USER ADDRESS METHOD
local all postgres peer
local all all scram-sha-256
hostssl app_db app_user 10.0.1.0/24 scram-sha-256
hostssl all dba_team 10.0.9.0/28 cert
hostssl replication repl 10.0.2.11/32 scram-sha-256
hostnossl all all 0.0.0.0/0 reject
# KALDIR ↓
host all all 0.0.0.0/0 trust
```

### UYGULA & DOĞRULA

```
-- Yeniden başlatmadan uygula
SELECT pg_reload_conf();

-- Kuralları ve hataları kontrol et
SELECT line_number, type, database,
       user_name, address, auth_method, error
FROM   pg_hba_file_rules
ORDER BY line_number;

-- md5 hash'li kullanıcıları bul
SELECT rolname FROM pg_authid
WHERE  rolpassword LIKE 'md5%';
```

⚠️ **Dikkat:** pg\_hba.conf **ilk eşleşen satırı** uygular — kısıtlayıcı kuralları üste yaz. Reload öncesi **error** kolonu boş olmalı; aksi halde kural yüklenmez.

## 02 Parola Politikası



| PARAMETRE / ÖZELLİK     | ÖNERİLEN      |
|-------------------------|---------------|
| password_encryption     | scram-sha-256 |
| passwordcheck           | Etkin         |
| VALID UNTIL             | 90 gün        |
| CREATEROLE              | Kısıtla       |
| LOGIN + NOINHERIT       | Servis hesabı |
| auth_delay.milliseconds | 500           |

### POSTGRESQL.CONF

```
password_encryption = 'scram-sha-256'
shared_preload_libraries = 'passwordcheck,auth_delay'
auth_delay.milliseconds = 500
authentication_timeout = 30s
```

### ROL & PAROLA YÖNETİMİ

```
-- İnsan hesabı: süreli parola
CREATE ROLE ali LOGIN PASSWORD '6%7xQ!r2...'
VALID UNTIL '2027-01-10';

-- Servis hesabı: miras yok, bağlantı limiti
CREATE ROLE svc_api LOGIN NOINHERIT
CONNECTION LIMIT 50 PASSWORD '...';

-- CREATEROLE yetkisini geri al
ALTER ROLE app_admin NOCREATEROLE;

-- Süresi dolan / dolacak parolalar
SELECT rolname, rolvaliduntil
FROM   pg_roles
WHERE  rolcanlogin
AND    rolvaliduntil < now() + interval '14 days';
```

⚠️ **Dikkat:** passwordcheck yalnızca düz metin gönderilen parolaları denetleyebilir. psql \password parolayı istemcide hash'ler — bu durumda kontrol atlanı.

✅ **İyi Pratik:** PG16+ ile CREATEROLE sahibi yalnızca ADMIN OPTION sahibi olduğu rolleri yönetebilir — yetki zincirini düzenli denetle.

## 03 Ağ Güvenliği postgresql.conf



| PARAMETRE                 | VARSAYILAN     | ÖNERİLEN               |
|---------------------------|----------------|------------------------|
| listen_addresses          | localhost      | 'localhost,10.0.1.5'   |
| port                      | 5432           | 5432 / özel            |
| ssl                       | off            | on                     |
| ssl_min_protocol_version  | TLSv1.2        | TLSv1.2+               |
| ssl_ciphers               | HIGH:MEDIUM... | HIGH:!aNULL:!MD5:!3DES |
| ssl_prefer_server_ciphers | on             | on                     |
| max_connections           | 100            | ihtiyaca göre          |

DoS yüzeyini sınırla; pooler (PgBouncer) kullan.

### SSL/TLS YAPILANDIRMASI

```
listen_addresses = 'localhost,10.0.1.5'
ssl              = on
ssl_cert_file    = 'server.crt'
ssl_key_file     = 'server.key' # chmod 600
ssl_ca_file      = 'root.crt'
ssl_min_protocol_version = 'TLSv1.2'
ssl_ciphers      = 'HIGH:!aNULL:!MD5:!3DES:!RC4'
```

### FIREWALL (KATMANLI SAVUNMA)

```
# Sadece uygulama alt ağı → 5432
ufw allow from 10.0.1.0/24 to any port 5432 proto tcp
ufw deny 5432/tcp

# firewallld alternatifi
firewall-cmd --permanent --add-rich-rule='rule family=ipv4
source address=10.0.1.0/24 port port=5432 protocol=tcp accept'
```

### AKTİF BAĞLANTILARDA SSL KONTROLÜ

```
SELECT a.username, a.client_addr, s.ssl, s.version, s.cipher
FROM   pg_stat_ssl s JOIN pg_stat_activity a USING (pid)
WHERE  a.backend_type = 'client backend';
```

✅ **İyi Pratik:** Firewall + listen\_addresses + pg\_hba.conf **üçlü savunma** katmanıdır. Biri hatalı yapılandırılrsa bile diğerleri korur. Veritabanını asla doğrudan internete açma.

04

Rol ve Yetki Yönetimi

| HAZIR ROL (PREDEFINED) | KULLANIM                                    |
|------------------------|---------------------------------------------|
| pg_read_all_data       | Tüm tablolarda SELECT (PG14+) — raporlama   |
| pg_write_all_data      | Tüm tablolarda INSERT/UPDATE/DELETE (PG14+) |
| pg_monitor             | İzleme araçları — superuser gerekmez        |
| pg_signal_backend      | Sorgu iptal / oturum sonlandırma            |
| pg_checkpoint          | CHECKPOINT çalıştırma (PG15+)               |
| pg_maintain            | VACUUM, ANALYZE, REINDEX (PG17+)            |

LEAST PRIVILEGE + PUBLIC ŞEMASI

```
-- public şemasını kilitle (PG15+ CREATE varsayılan kapalı)
REVOKE ALL ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE app_db FROM PUBLIC;

-- Grup rolü → kullanıcıya ata
CREATE ROLE app_ro NOLOGIN;
GRANT CONNECT ON DATABASE app_db TO app_ro;
GRANT USAGE ON SCHEMA app TO app_ro;
GRANT SELECT ON ALL TABLES IN SCHEMA app TO app_ro;
ALTER DEFAULT PRIVILEGES IN SCHEMA app
  GRANT SELECT ON TABLES TO app_ro;
GRANT app_ro TO report_user;
```

SUPERUSER DENETİMİ & SET ROLE

```
SELECT rolname, rolsuper, rolcreatorole,
       rolcreatedb, rolbypassrls, rolcanlogin
FROM   pg_roles
WHERE  rolsuper OR rolcreatorole OR rolbypassrls;

-- Geçici yetki yükseltme (NOINHERIT kullanıcı)
SET ROLE dba_admin; -- işlemi yap
RESET ROLE;
```

ROW-LEVEL SECURITY (RLS)

```
ALTER TABLE orders ENABLE ROW LEVEL SECURITY;
ALTER TABLE orders FORCE ROW LEVEL SECURITY;

CREATE POLICY tenant_isolation ON orders
  USING (tenant_id = current_setting('app.tenant_id')::int);
```

⚠️ ⚠️ Dikkat:

 Tablo sahibi ve BYPASSRLS rolleri RLS'yi atlar. Sahip için FORCE ROW LEVEL SECURITY kullan; uygulama asla tablo sahibi rolüyle bağlanmamalı.

05

Denetim ve Loglama

| PARAMETRE                  | DEĞER | AÇIKLAMA                                 |
|----------------------------|-------|------------------------------------------|
| log_connections            | on    | Başarılı/başarısız bağlantıları logla    |
| log_disconnections         | on    | Oturum süresi ile çıkışları logla        |
| log_statement              | 'ddl' | 'mod' = DDL + DML yazma işlemleri        |
| log_min_duration_statement | 1000  | 1 sn üzeri yavaş sorgular (ms)           |
| log_lock_waits             | on    | deadlock_timeout aşan kilit beklemeleri  |
| log_file_mode              | 0600  | Log dosyalarını sadece sahibi okusun     |
| log_parameter_max_length   | 0     | Bind parametrelerini (hassas veri) gizle |

ÖNERİLEN LOG\_LINE\_PREFIX

```
log_line_prefix =
'%t [%p]: [%L-1] user=%u,db=%d,app=%a,client=%h '
# %t zaman · %p PID · %L satır no · %u kullanıcı
# %d veritabanı · %a uygulama · %h istemci IP
```

PGAUDIT — DETAYLI AUDIT TRAIL

```
# postgresql.conf (restart gerekir)
shared_preload_libraries = 'pgaudit'
pgaudit.log              = 'ddl, role, write'
pgaudit.log_catalog      = off
pgaudit.log_relation     = on
pgaudit.log_parameter    = off      # hassas veri sızmasın
pgaudit.role             = 'auditor'
```

```
CREATE EXTENSION pgaudit;

-- Object-level audit: hassas tablo
CREATE ROLE auditor NOLOGIN;
GRANT SELECT, UPDATE ON customers_pii TO auditor;

-- Sadece belirli kullanıcı için ek audit
ALTER ROLE contractor SET pgaudit.log = 'read, write';
```

ÖRNEK AUDIT LOG SATIRI

```
AUDIT: SESSION,1,1,DDL,DROP TABLE,public.orders,
      DROP TABLE orders;<not logged>
```

⚠️ ⚠️ Dikkat:

 log\_statement = 'all' ve pgaudit.log = 'all' production'da disk/I/O patlamasına yol açar ve parolaları loglara yazabilir. Logları merkezi SIEM'e gönder.

06

Şifreleme

| KATMAN             | YÖNTEM                            |
|--------------------|-----------------------------------|
| Aktarımdaki veri   | ssl=on + sslmode=verify-full      |
| Kolon seviyesi     | pgcrypto (pgp_sym_encrypt)        |
| Durağan veri (TDE) | pg_tde (PG17 · Percona)           |
| Disk seviyesi      | LUKS / dm-crypt / cloud KMS       |
| Yedekler           | pg_dump   gpg · pgBackRest cipher |

PGCRYPTO — KOLON ŞİFRELEME

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;

-- Simetrik şifreleme (bytea kolon)
INSERT INTO customers (name, tckn_enc)
VALUES ('Ayşe', pgp_sym_encrypt('12345678901', :'key'));

SELECT pgp_sym_decrypt(tckn_enc, :'key') FROM customers;

-- Uygulama parolası: bcrypt hash
SELECT crypt('s3cret', gen_salt('bf', 12));
```

PG\_TDE — TRANSPARENT DATA ENCRYPTION

```
shared_preload_libraries = 'pg_tde'
CREATE EXTENSION pg_tde;
-- Key provider + principal key tanımla, sonra:
CREATE TABLE cards (...) USING tde_heap;
```

ŞİFRELİ BACKUP

```
pg_dump -Fc app_db | gpg --encrypt \
  --recipient dba@corp.com > app_db.dump.gpg

gpg --decrypt app_db.dump.gpg | pg_restore -d app_db

# pgBackRest: repo1-cipher-type=aes-256-cbc
```

SERTİFİKA & İSTEMCİ DOĞRULAMA

```
ALTER SYSTEM SET ssl_cert_file = 'server.crt';
ALTER SYSTEM SET ssl_key_file  = 'server.key';
SELECT pg_reload_conf();

# İstemci: MITM'e karşı tam doğrulama
psql "host=db.corp.com dbname=app_db user=app_user
      sslmode=verify-full sslrootcert=root.crt"
```

⚠️ ⚠️ Dikkat:

 pgcrypto anahtarını SQL'e gömmе — log\_statement ile loglara düşebilir. sslmode=require sertifikayı doğrulamaz; verify-full kullan. Community PostgreSQL'de yerleşik TDE yoktur.

07

Extension Güvenliği

| Extension             | Amaç                                     | Risk    |
|-----------------------|------------------------------------------|---------|
| pgaudit               | Audit trail                              | GÜVENLİ |
| pgcrypto              | Kriptografik fonksiyonlar                | GÜVENLİ |
| pg_stat_statements    | Sorgu istatistikleri                     | GÜVENLİ |
| postgres_fdw / dblink | Uzak bağlantı — kimlik bilgisi sızıntısı | DİKKAT  |
| adminpack / file_fdw  | Sunucu dosya sistemine erişim            | YÜKSEK  |
| plpython3u / plperl   | Untrusted dil — OS komutu çalıştırır     | YÜKSEK  |

SHARED\_PRELOAD\_LIBRARIES WHITELIST

# Sadece onaylı kütüphaneler — restart gerekir  
shared\_preload\_libraries =  
    'pg\_stat\_statements,pgaudit,passwordcheck'  
# Mevcut değer: **SHOW** shared\_preload\_libraries;

KURULUM & FONKSİYON ERIŞİMİ

-- Yetki kontrolü: extension'ı dedicated şemaya kur  
CREATE SCHEMA ext AUTHORIZATION postgres;  
CREATE EXTENSION IF NOT EXISTS pgcrypto SCHEMA ext;  
  
-- Fonksiyonlar varsayılan olarak PUBLIC'e açıktır!  
REVOKE EXECUTE ON ALL FUNCTIONS IN SCHEMA ext FROM PUBLIC;  
GRANT EXECUTE ON FUNCTION ext.pgp\_sym\_decrypt(bytea, text)  
    TO app\_secure;  
ALTER DEFAULT PRIVILEGES  
    REVOKE EXECUTE ON FUNCTIONS FROM PUBLIC;  
  
-- Kurulu extension envanteri  
SELECT e.extname, e.extversion, n.nspname AS schema,  
        r.relname AS owner  
FROM   pg\_extension e  
JOIN   pg\_namespace n ON n.oid = e.extnamespace  
JOIN   pg\_roles r     ON r.oid = e.extowner;

SECURITY DEFINER FONKSİYONLAR

CREATE FUNCTION app.rotate\_key() RETURNS void  
    LANGUAGE plpgsql SECURITY DEFINER  
    SET search\_path = pg\_catalog, pg\_temp -- zorunlu!  
AS \$\$ ... \$\$;

Truste extensions (PG13+): trusted = true olan extension'lar, veritabanında CREATE yetkisi olan superuser olmayan kullanıcılarca kurulabilir. Daha sıkı kontrol için pgextwList ile whitelist uygula.

⚠️ Dikkat:

Untrusted diller (\*u) ve COPY ... PROGRAM superuser'a OS seviyesinde komut çalıştırma imkânı verir — bu yüzden superuser = OS kullanıcısı postgres demektir.

08

Güvenlik Kontrol Listesi

GO-LIVE ÖNCESİ

| Kontrol Maddesi                                                                                                      | Öncelik  |
|----------------------------------------------------------------------------------------------------------------------|----------|
| <div>✓</div> <div>pg_hba.conf — trust kaldırıldı<br/>pg_hba_file_rules WHERE auth_method='trust'</div>               | KRİTİK   |
| <div>✓</div> <div>scram-sha-256 aktif, md5 hash kalmadı<br/>SHOW password_encryption;</div>                          | KRİTİK   |
| <div>✓</div> <div>SSL/TLS etkin, TLSv1.2+ zorunlu<br/>SHOW ssl; · pg_stat_ssl</div>                                  | KRİTİK   |
| <div>✓</div> <div>Superuser sayısı ≤ 2<br/>SELECT count(*) FROM pg_roles WHERE rolsuper;</div>                       | KRİTİK   |
| <div>✓</div> <div>public şeması kısıtlandı<br/>\dn+ public</div>                                                     | ÖNEMLİ   |
| <div>✓</div> <div>pgaudit kurulu ve aktif<br/>SHOW pgaudit.log;</div>                                                | ÖNEMLİ   |
| <div>✓</div> <div>Parola politikası uygulandı (passwordcheck + VALID UNTIL)<br/>SHOW shared_preload_libraries;</div> | ÖNEMLİ   |
| <div>✓</div> <div>Backup şifrelemesi aktif<br/>pgg / repo1-cipher-type</div>                                         | ÖNEMLİ   |
| <div>✓</div> <div>RLS kritik tablolarda etkin<br/>pg_class WHERE relrowsecurity</div>                                | ÖNERİLEN |
| <div>✓</div> <div>log_connections açık<br/>SHOW log_connections;</div>                                               | ÖNERİLEN |

KRİTİK

hemen düzelt

ÖNEMLİ

bu sprint

ÖNERİLEN

planla

TEK SORGUDA GÜVENLİK DENETİMİ

-- dbaguide.com · security quick audit  
SELECT name, setting,  
    CASE  
        WHEN name = 'ssl' AND setting = 'on'       THEN '✓'  
        WHEN name = 'password\_encryption'  
            AND setting = 'scram-sha-256'       THEN '✓'  
        WHEN name IN ('log\_connections',  
                      'log\_disconnections')  
            AND setting = 'on'                THEN '✓'  
        WHEN name = 'shared\_preload\_libraries'  
            AND setting LIKE '%pgaudit%'       THEN '✓'  
        WHEN name = 'listen\_addresses'  
            AND setting <> '\*'                THEN '✓'  
        ELSE 'x KONTROL ET'  
    END AS status  
FROM   pg\_settings  
WHERE  name IN ('ssl', 'password\_encryption',  
                'log\_connections', 'log\_disconnections',  
                'shared\_preload\_libraries', 'listen\_addresses');  
  
-- RLS kapalı kritik tablolar  
SELECT relname FROM pg\_class  
WHERE  relkind = 'r' AND NOT relrowsecurity  
AND   relnamespace = 'app':regnamespace;

✓ İyi Pratik:

Bu kontrolleri CI/CD veya cron ile haftalık çalıştır, sonuçları izleme sistemine gönder. Her major upgrade sonrası listeyi baştan uygula.