Melhores Práticas PostgreSQL

Versão: v1
Criado por: Rafael Domiciano
Disponibilidade: Pública

Essa listagem contém boas-práticas de configuração e gestão de banco de dados Postgres da ótica de segurança. Essa listagem não compreende todos as ações que podem ser tomadas, mas com a devida aplicação delas terá resultados rápidos na proteção do seu database.

Faça hoje mesmo

Ações que podem ser realizadas de forma rápida que já trarão o mínimo de proteção para o seu banco de dados.

- Troque a senha dos seus usuários do banco
- Crie usuários de aplicaçoes e altere suas aplicações para autenticar com esse novo usuário
- Crie um backup e salve num local somente-leitura (como um bucket na cloud de preferência)
- Retire autenticações trust do seu pg_hba.conf. Cuidado, isso pode quebrar alguma aplicação que estejam mandando senha inválidas (ou nem mandando senha para autenticação)
- Caso seu banco de dados esteja público apenas para que desenvolvedores acessem, feche hoje mesmo e coloque um túnel SSH/bastion para esse acesso. Esse é um processo rápido de ser feito, e as ferramentas de acesso a banco de dados, em sua maioria, dispõem de configuração de túenl SSH. Não coloque seu ambiente em risco por conta disso
    - Caso não possa realizar essa ação, feche os acessos (por exemplo: em AWS, faça no security group) para os IP's das pessoas que precisam utilizar

Faça com tempo

Ações que podem exigir um pouco de tempo para aplicação, janelas e estudos internos. Não deixe de realizar por conta disso. Cada item abaixo é 1 camada adicional de proteção para o seu banco de dados. Note que muitas delas vale para outros SGDB, portanto amplie seu foco de atuação e tenha uma infra mais segura.

- Senha aleátoria. Troca de senha periodica
    - De prefêrencia para utilização de um cofre de senha (ex: AWS Secrets Manager, Hashicorp Vault, Boundary). A leitura da senha pode ser realizado 1 única vez na inicialização da aplicação, que ficará em memória. Em caso de troca de senha é um processo mais rápido de ser feito com menor impacto no seu negócio. Outro benefício com essa prática que as senhas não ficarão em arquivos de configuração espelhados pela sua estrutura
- Utilize um banco de dados diferente do "postgres"
- Utilize usuários de gestão/DBA para o banco de dados. Defina uma senha realmente complexa para o usuário "postgres" e reserve sua utilização para um último caso

- Utilize uma porta diferente da 5432. A porta 5432 (PG) é padrão e será a primeira a ser procurada na sua rede
- Faça backups periodicos (Diário se possivel) e salve num local somente-leitura, como um bucket S3 (habilite versionamento e "deny-delete")
    - Configure PITR
- NÃO faça conexões ao banco de dados diretamente da sua aplicação frontend. Utilize um backend para isso
- Faça testes de SQL Injection na sua aplicação (conta também para aplicações mobile)
    - Lembre-se que com SQL Injection é possível dropar e extrair dados
- Crie roles específicas para usuários e aplicações
    - Traga para dentro do banco de dados o conceito de PLP (Principle of Least Privilege - Princípio do menor privilegio)
    - Por exemplo:
        - Aplicações de BI podem ser somente-leitura, nas tabelas específicas
        - backend somente SELECT, INSERT, UPDATE
- Não deixe seu banco aberto para parceiros ou terceiros sem a real necessidade. A rede deles pode ser impactada, e mesmo numa VPN, o malware/ransonware pode transbordar para a sua. Esteja alinhado com o time de segurança nesse item
    - Essa também pode ser uma fonte de vazamento de dados
- Utilize conexões com SSL, para a senha não trafegar em plain-text na rede
- pg_hba.conf: elimine nesse momento do seu conhecimento o metódo de autenticação "trust", mesmo para conexões locais. Esse método não faz validação de senha, ou seja, qualquer um pode acessar seu banco se acertar o usuário. Elimine do pg_hba.conf também
- Verifique o log do banco em busca de sugestões de conexões que pareçam suspeitas
    - Habilite (temporariamente) a configuração "log_connections"
- Banco exposto em rede pública nunca mais. Sempre faça deploys na nuvem em subnets privadas
    - Utilize um túnel ssh/bastion, ou um proxy de banco caso necessário
    - Seu banco precisa estar em subnet pública por algum motivo: Feche o máximo possível seu security group

Replicação

- Se dispor de recursos (financeires e hardware) tenha uma replicação do seu banco de dados
- A replicação é 1 camada a mais de proteção, ela não elimina um backup com pg_dump

Correções de segurança

O Postgres implementa correções de segurança em todas as versões que estão suportadas. Na data de hoje (20/09/2023) está suportado as versões 11 para frente. Você pode verificar nesse site (https://www.postgresql.org/support/versioning/) mais detalhes sobre os releases, data provável do último release e se está suportado ou não.
Uma boa-prática é estar sempre na versão "minor" mais recente da versão que está sendo utilizada.