16.326 bancos de dados abertos, e a correção são duas linhas de SQL

Análise semanal LATAMSEC · semana de 22 a 29 de setembro de 2026 · exposição sistêmica em aplicações Supabase

Resumo executivo. A UpGuard identificou 16.326 bancos Supabase com tabelas legíveis por qualquer visitante, a partir de uma varredura em cerca de 300 mil domínios. Mais da metade apresentava indício de dado pessoal; um subconjunto trazia senha e token de autenticação em texto claro. Não houve exploração de vulnerabilidade: a chave pública do projeto devolve a tabela inteira quando a política de segurança por linha não foi escrita. O padrão se concentra no caminho programático de criação de tabela, que é o caminho usado por agentes de codificação.

1. O que foi encontrado

Greg Pollock, diretor de pesquisa da UpGuard, publicou em 25 de setembro de 2026 o levantamento que deu origem a esta análise. A metodologia partiu da impressão digital técnica de uso do Supabase em arquivos JavaScript públicos, cruzando dados do BuiltWith com consultas ao Chrome UX Report. Dada a escala, os pesquisadores consultaram o esquema das tabelas em vez de ler os registros, o que significa que o número de pessoas afetadas é maior do que o relatório demonstra.

Indicador	Valor
Domínios com indício de uso do Supabase	Cerca de 300 mil
Bancos com tabelas legíveis sem autenticação	16.326
Bases com indício de dado pessoal	Mais da metade
Bases com senha ou token de autenticação	Subconjunto menor, com casos em texto claro
Bases criadas recentemente com apoio de IA	Mais de 60%, segundo a UpGuard

2. Por que o erro se repete

O Supabase expõe uma API REST sobre o Postgres e entrega, com o projeto, uma chave anônima embutida no JavaScript do site. Ela é pública por desenho e está documentada como tal. O controle de acesso é a política de segurança por linha, o Row Level Security, que precisa ser escrita tabela por tabela. Sem ela, a chave pública devolve a tabela inteira.

Desde março de 2025, após a CVE-2025-48757, a proteção vem habilitada por padrão nas tabelas criadas pelo editor visual. Tabela criada por API não recebe a proteção por padrão, e criar tabela por API é exatamente como um agente de codificação opera. O caminho que ficou seguro é o do humano clicando; o que ficou aberto é o da automação.

> A lição não é sobre Supabase. Quando uma plataforma corrige um padrão inseguro apenas na interface visual, ela protege quem já sabia o que estava fazendo e deixa descoberto quem delegou a decisão. O agente não escreve a política que ninguém pediu.

3. O que estava legível

Aplicação	Dado acessível
Serviço de manobrista, nordeste dos EUA	Mais de 100 mil clientes com telefone, cerca de 43 mil com nome e e-mail, cerca de 78 mil placas e histórico de visitas
Serviço de senha de uso único, Filipinas	Mais de 2 mil cadastros e mais de 100 mil mensagens SMS contendo os próprios códigos de verificação
Consultoria de mudança, Canadá	Quase 5 mil registros, dos quais 884 com senha em texto claro
Consulado de país africano	25 mil cadastros com endereço residencial e local de acolhimento em emergência
Plataforma de criadores, Índia	Mais de 100 mil mensagens privadas e dado de pagamento

O caso das senhas de uso único é o mais instrutivo. Uma base que armazena o código enviado por SMS e o devolve a qualquer requisição transforma o segundo fator em formalidade.

4. Roteiro de noventa dias

Dias 1 a 30 — descobrir

- Levante todas as aplicações web da organização, incluindo as que não estão sob o time de tecnologia. Comece pelos domínios e subdomínios registrados, não pela lista de sistemas.
- Procure no código-fonte público de cada página por indício de chave e endereço de projeto Supabase, Firebase ou equivalente.
- Registre quem construiu cada aplicação e com qual ferramenta. A resposta "foi a agência" é marcador de prioridade, não encerramento.

Dias 30 a 60 — testar

- Com a chave anônima obtida do próprio site, consulte as tabelas principais em ambiente controlado e com autorização formal.
- Verifique especificamente tabelas de usuário, mensagem, pedido e pagamento. São as mais encontradas abertas.
- Documente o resultado por aplicação. O registro do teste sustenta a decisão posterior de notificar.

Dias 60 a 90 — corrigir e institucionalizar

- Habilite a segurança por linha e escreva a política em toda tabela que responda à chave pública.
- Inclua a verificação no roteiro de publicação, independentemente de quem escreveu a aplicação.
- Trate criação de tabela por API como caminho que exige revisão explícita.

5. A leitura de LGPD

Base com dado pessoal legível sem autenticação é acesso não autorizado a dado pessoal. O relógio da comunicação começa quando a organização toma conhecimento, o que inclui o resultado do teste descrito acima. Descobrir e não registrar não pausa nada.

A responsabilidade não migra para a agência que construiu a aplicação: o controlador é quem determina a finalidade do tratamento. E senha armazenada em texto claro é falha de segurança anterior ao incidente, com peso próprio em eventual fiscalização.

6. Três evidências para o conselho

Evidência	O que ela prova
Inventário de aplicação por origem	Que a organização sabe quais aplicações web existem, quem as construiu e com qual ferramenta. Sem ela, nenhuma outra evidência é verificável.
Resultado do teste como visitante	Que a política de acesso existe e funciona, com o registro da consulta feita com a chave pública e do que ela devolveu.
Checagem no roteiro de publicação	Que a verificação virou etapa obrigatória antes da produção, e não uma ação pontual de campanha.

7. Leitura LATAMSEC

O que essa pesquisa mede não é a qualidade de um produto. É o que acontece com a segurança quando a velocidade de construção sobe e a revisão não acompanha. Construir uma aplicação com banco, autenticação e publicação levava semanas e passava por pelo menos uma pessoa que sabia o que era política de acesso. Hoje leva uma tarde e pode não passar por ninguém.

A pergunta que vale para o próximo trimestre não é sobre Supabase. É quantas aplicações a organização colocou no ar nos últimos doze meses sem que o time de segurança soubesse, e quantas delas guardam dado de pessoa real.

Fontes

- UpGuard — Everything Everywhere: systemic data exposure in Supabase apps (Greg Pollock, 25/09/2026).
- Supabase — documentação oficial de Row Level Security.
- Unite.AI — UpGuard study finds 16,326 Supabase databases exposing readable tables.
- OWASP — A01:2021 Broken Access Control.