UUID para melhorar a segurança dos SGBDs

Walter Nascimento Barroso

Abstract—This article introduces a change of ID to UUID. Specifically, to prevent the security flaw that a sequential ID provides. In this case, the problem is known a security fault from which from an ID it is possible to discover the following sequences. The solution presented is applied directly to PostgreSQL, replacing the IDs with UUIDs. The goal is to improve the security of existing databases.

Keywords—UUID, SGBD, PostreSQL, data bases.

1 Introdução

TODO desenvolvedor aprende desde cedo a usar ID sequencial para definir a primary key de suas tabelas, facilitando assim na hora de fazer uma busca, ou uma manipulação de algum registro, mas o ID sequencial não tem apenas vantagens, também possui desvantagens e uma delas facilita pessoas mal intencionadas a encontrar informações do qual não devia ser possivel encontrar.

Ao possuir o ID senquencial, é facil encontrar o seguinte, simplesmente somando mais um ao valor encontrado ou diminuindo um em um sistema mal configurado é possivel saber os valores dos proximos registros.

2 O que é UUID?

UUID é a sigla em ingês para Universally unique identifier(Identificador Único Universal). O UUID foi padronizado pela Open Software Foundation como um identificador padrão para softwares. O objetivo é criar um idenficador único que possa ser compartilhado com outros softwares, facilitando a troca de informações entre sistemas.

A definição para um UUID é um número de 128 bits. Em teoria, o número possível de UUIDs geradas é de 3×10^{38}.

3 Casos de Problemas com ID sequências

Alguns empresas famosas ja sofreram por usar os indentificadores sequenciais, uma delas foi a Bematech, uma empresa de grande porte e referencia na area de automação, dentro do site http://www.bematechmais.com.br/, antes era possivel atraves de força bruta, inserindo varios numeros aleatorios na pagina de cadastro, em um certo momento dava exatamente a um registro na base e com isso era possivel conferir toda a lista de empresas parceiras e revendedores que a companhia possui.

Outra empresa que foi alvo de mal uso dos identificadores sequenciais foi a trampos.co um site de anuncio de vagas, muito utilizado principalmente por startups. um dos sites mais acessados, e neste site todos a maioria dos usuarios tem cartoes de creditos validos, e os cadastro são realizados por identificadores sequenciais e a algum tempo essa falha foi explorada, o especialista que verificou a falha fez alguns teste e posteriormente informou a empresa, mas não é todas as empresas que tem essa sorte.

• *Walter Nascimento Barroso, Analista de Sistemas, E-mail: walternascimentobarroso@gmail.com*

4 Por que usar UUID?

UUID é além de ser simples de usar, ele ja proteje a aplicação em caso de ID sequenciais. Como foi mostrado na seção de Casos de Problemas com ID sequências, a falha existe e muitos usuarios gostam de verificar a confiabilidade do site, o UUID fornece um hash que a primeira vista é bem complexo de intender e para o usuario comun dificil de manipula, mas a principal vantagem é a impossibilidade de descobrir os indetificadores dos outros registros.

5 Implementando UUID no PostgreSQL

O PostgreSQL já possui um tipo de dado para UUID, então para usar como identificador basta seguir a SQL abaixo:

Tabela 1
Usuario

Linha	Script
1	CREATE TABLE usuario (
2	id uuid primary key,
3	nome varchar(50),
4	idade int
5	);

Com este script a tabela é criada usando o UUID como chave primaria, o PostgreSQL por padrão já tem o tipo UUID como tipo de dado, mas para gerar UUID por ser uma sequencia aleatoria, é necessario habilitar uma extenção, segue abaixo o script:

Tabela 2
Habilitando UUID

Linha	Script
1	CREATE EXTENSION IF NOT EXISTS ”uuid-ossp”;
2	SELECT uuid_generate_v4();

6 Conclusão

ID sequencial de forma mal implementa é uma falha de segurança, e para remover este problema UUID é uma excelente solução, fácil implementação e uso.

UUID de inicio é algo bem robusto, mas com essa robustez ele traz mais qualidade e mais segurança a aplicação.

A maioria dos SGBDs ja trazem UUID como um tipo de dado padrão e as principais linguagens de programação ja fornecem funções praticas para geração de UUID.