Tipos de Encriptación en la Función password_hash()

Introducción

Al trabajar con contraseñas en aplicaciones, es fundamental almacenarlas de manera segura. La función password_hash() de PHP permite protegerlas transformándolas en un hash, un código irreconocible que no puede revertirse a su forma original.

¿Cómo Funciona la Encriptación?

Transformación: La contraseña se convierte en un hash único mediante un algoritmo.

Uso de "sal": Se añade un valor aleatorio para que incluso contraseñas iguales generen hashes distintos.

Dificultad ajustable: Los algoritmos modernos permiten ajustar la complejidad del cálculo para hacerlo más seguro frente a ataques.

Al iniciar sesión, la contraseña ingresada se compara con el hash almacenado mediante la función password_verify().

Algoritmos de Encriptación en password_hash()

1. BCRYPT

Un algoritmo clásico que transforma la contraseña en un hash seguro. Utiliza rondas repetitivas de cálculo para complicar el proceso.

Ventajas: Fiable, ampliamente compatible y seguro por más de dos décadas.

Limitaciones: Solo soporta contraseñas de hasta 72 caracteres.

Ideal para sistemas que necesitan compatibilidad con software antiguo.

2. ARGON2

El algoritmo más avanzado y seguro disponible hoy en día. Fue diseñado para aprovechar las capacidades de los ordenadores modernos y resistir ataques más sofisticados.

Tipos:

ARGON2i: Resiste ataques basados en tiempo.

ARGON2d: Previene ataques de canal lateral (menos usado para contraseñas).

ARGON2id: Combina las ventajas de los dos anteriores (recomendado).

Limitaciones: Es más lento y consume más recursos, pero esto lo hace más resistente a ataques de fuerza bruta.

Más seguro y flexible que BCRYPT. Permite ajustar no solo las rondas de cálculo, sino también el uso de memoria y hilos de procesamiento.

Para proyectos modernos que priorizan la seguridad.

¿Cómo se Protegen las Contraseñas en PHP?

Al registrarse:

La contraseña del usuario se transforma en un hash único con password_hash() y se almacena en la base de datos.

Al iniciar sesión:

La contraseña ingresada se verifica comparándola con el hash almacenado usando password_verify(). Si coinciden, el usuario accede.

Esto asegura que las contraseñas nunca se almacenen ni viajen en texto visible.