Parolalar neden veritabanında düz metin olarak saklanmamalıdır?



www.yzokul.com.tr

Parolalar neden veritabanında düz metin olarak saklanmamalıdır?



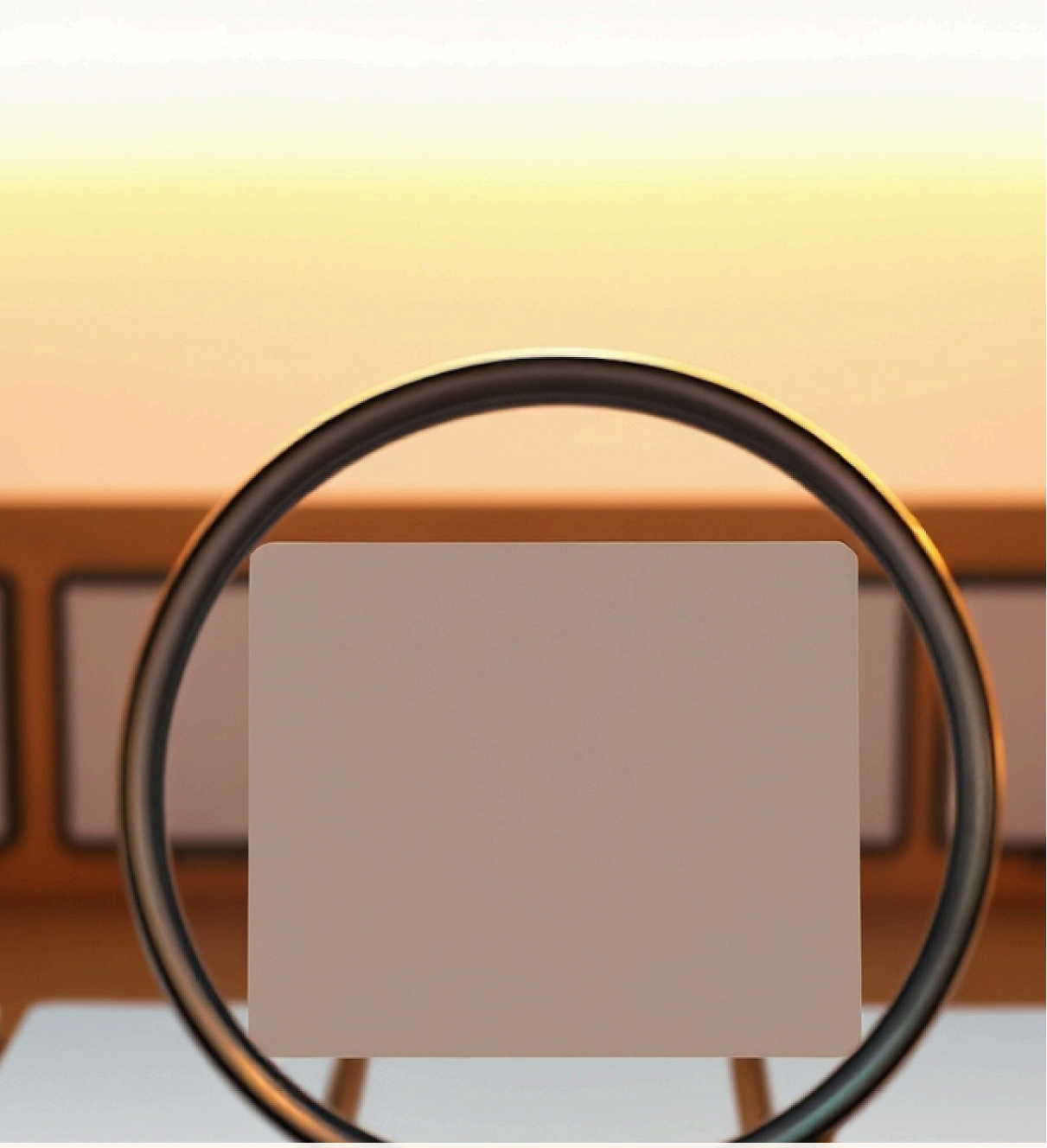
Şu soruyu görüyorsun: parolalar neden bir veritabanında düz metin olarak saklanmamalıdır? Kısa yanıt, kayıtları ele geçiren herkesin parolaları kolayca okuyabileceğidir. Bu, kayıtlar alındığı anda saklanan her parolanın açığa çıkacağı anlamına gelir. Bu yüzden parolaları okunabilir bırakmak yerine korumamız gerekir. Bu kitap, bu korumanın nasıl işlediğini tamamen yeni başlayanlar için adım adım açıklar.



Bir veritabanının, kullanıcı hesapları gibi bir sistem tarafından tutulan kayıtlar topluluğu olduğunu anlıyorsun. Her kayıt, kullanıcı adı ve parola gibi etiketlenmiş veri parçaları olan alanlar içerebilir. Veritabanı, sistemin bu bilgileri daha sonra kontrol edebilmek için sakladığı yerdir. Bir hesap oluşturduğunda, bilgilerinle birlikte yeni bir kayıt eklenir. Parola alanı, burada en çok önemsediğimiz kısımdır.



Düz metnin, parolanın kullanıcının yazdığı şekilde, hiçbir değişiklik yapılmadan yazılması anlamına geldiğini görüyorsun. Bu şekilde saklamak, kaydın okunabilir parolanın kendisini harf harf içermesi demektir. Hiçbir karıştırma, kodlama veya herhangi bir dönüşüm yoktur. Kayda bakan herkes parolayı kullanıcının yazacağı gibi görür. Bu, onu saklamanın mümkün olan en basit ve en tehlikeli yoludur.



Kayıtları ele geçiren birinin onları açıp parolaları doğrudan okuduğunu hayal ediyorsun. Özel bir beceri gerekmez çünkü parolalar zaten okunabilir metindir. Tahmin etmeleri, çözmeleri veya herhangi bir araç kullanmaları gerekmez; sadece bakarlar. Düz metin riski budur: sır ortada durur. Kayıtlara sahip bir yeni başlayan bile her parolayı saniyeler içinde okuyabilir.



Kayıtların tek bir kez ele geçirilmesinin, saklanan her parolayı aynı anda açığa çıkardığını anlıyorsun. Tek bir sızıntı, yalnızca bir kişiyi değil, o veritabanında hesabı olan herkesi vurur. Etkilenen her kullanıcının hesabı artık kayıtları elinde tutan herkesin kötüye kullanımına açıktır. Zarar tek bir olayda tüm kullanıcılara yayılır. Bu yüzden parolaların nasıl saklandığı yalnızca bir hesap için değil, herkes için bu kadar önemlidir.



Birçok kişinin parolaları yeniden kullandığını, aynı parolayı başka hizmetlerde de kullandığını görüyorsun. Bu nedenle bir sistemden okunabilir bir parola, başka yerlerdeki hesapları da riske atabilir. Burada onu okuyan biri, e-postada, sosyal medyada veya alışveriş sitelerinde deneyebilir. Zarar, ele geçirilen sistemle sınırlı kalmaz. Bir sitedeki parola güvenliği, kullanıcının diğer hesapları için de önemlidir.



Amacı anlıyorsun: sistemin orijinal parolayı hiç geri okuması gerekmemelidir. Yalnızca girilen parolanın belirlenenle eşleşip eşleşmediğini kontrol etmesi yeterlidir. Sistem parolayı hiçbir zaman göstermek veya birine geri göndermek zorunda değildir. Bu tasarımı değiştirir: okuyabileceğin bir şeyi değil, kontrol etmeni sağlayan bir şeyi sakla. Bu fikir, daha güvenli saklamanın anahtarıdır.



Bir karma yönteminin, parolayı tek yönlü bir süreçle sabit görünümlü bir değere dönüştürdüğünü görüyorsun. O değerden orijinal parolanın geri elde edilmesi amaçlanmaz. Aynı girdi, aynı yöntemle her zaman aynı çıktıyı verir; kontrolü mümkün kılan da budur. Karma, şifreleme değildir çünkü onu geri çevirecek bir anahtar yoktur. O, parolanın kendisi değil, parolanın bir parmak izidir.



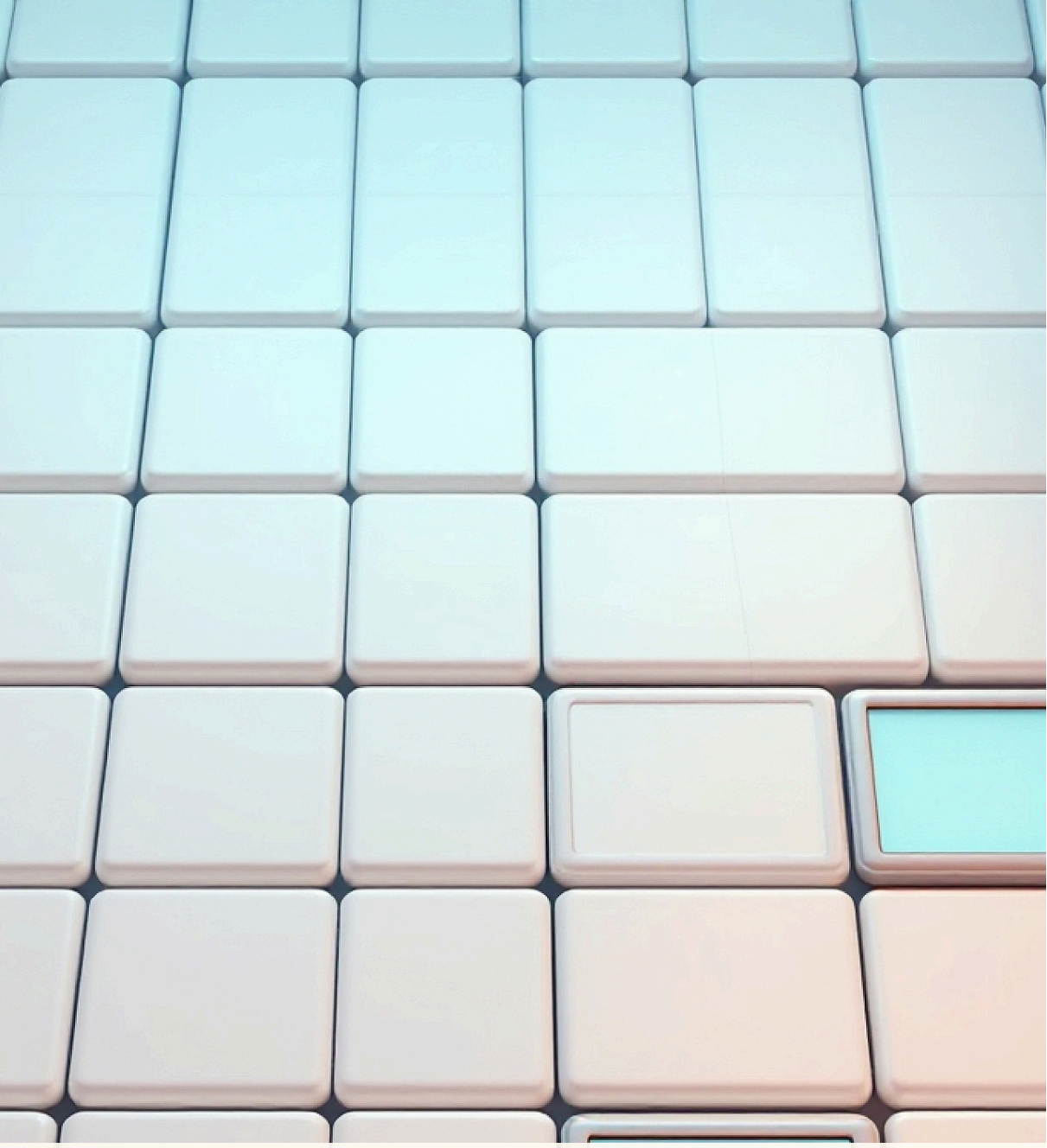
Veritabanı kaydının artık okunabilir parola yerine karma değeri tuttuğunu hayal ediyorsun. Kayıtları ele geçiren biri parolayı değil, yalnızca karmayı görür. Karma, belirgin bir anlamı olmayan karışık karakterler gibi görünür. Onu okumak, saldırgana kullanıcının ne yazdığını söylemez. Parola karmasının özü budur: işlenmiş değeri sakla, asla orijinali saklama.



Bir oturum açmayı izliyorsun: kullanıcı bir parola yazar, sistem onu aynı karma yöntemiyle işler ve sonucu saklanan değerle karşılaştırır. Eşleşme parolanın doğru olduğu, eşleşmeme ise yanlış olduğu anlamına gelir. Sistem orijinal parolayı hiç geri okumaz; yalnızca işlenmiş değerleri karşılaştırır. Bir sistem, parolayı hiçbir zaman okunabilir biçimde saklamadan bu şekilde kontrol eder.



Tuzun, karma işleminden önce karıştırılan ek veri olduğunu anlıyorsun. İki kullanıcı aynı parolayı seçse bile sonuç değerini değiştirir. Tuz, karma değerinin yanında saklanır ki sistem oturum açmada aynı süreci tekrarlayabilsin. Gizli olması gerekmez; işi her sonucu farklı kılmaktır. Tuz eklemek, korumayı büyük ölçüde iyileştiren basit bir adımdır.



Her kaydın kendine ait farklı bir tuz değeri aldığını görüyorsun. Aynı parolalar o zaman farklı saklanan değerler üretir, böylece tek bir tahmin birçok kaydı açmaz. Bu olmadan, aynı parolaya sahip iki kullanıcının karmaları aynı olurdu ve bir saldırgan bunları anında fark edebilirdi. Her kayıt için farklı bir tuz bu deseni bozar. Bu yüzden tuz değerleri kayıtlar arasında asla paylaşılmamalıdır.



Basit ve hızlı bir karmanın tek başına yeterli koruma sağlamayabileceğini öğreniyorsun. Hızlı işleme, bir saldırganın ele geçirilmiş bir karmaya karşı hızla çok sayıda tahmin denemesine olanak tanır. Hız sistem için yararlıdır ama saldırgana da yardım eder. Bu yüzden parola karması, yavaş ve tahmin edilmesi maliyetli olacak şekilde tasarlanmış yöntemler kullanmalıdır. Yalnızca hızlı karmalar parola güvenliği için yeterli değildir.



Parolaları okunabilir tutma gibi yaygın hatayı, belki de veritabanının yeterince özel olduğunu düşünerek fark ediyorsun. Kayıtlar yine de ele geçirilebilir ve o zaman parolalar açığa çıkar. Gizlilik ayarları, saklanan metnin okunabilir olduğu gerçeğini değiştirmez. Veritabanının güvenli olduğunu varsaymak bir koruma değildir. Çözüm, okunabilir parolaları saklamayı bırakıp onları karma işleminden geçirmektir.



Her kayıt için farklı bir tuz olmadan karma işlemi yapma hatasını fark ediyorsun. O zaman eşit parolalar depoda eşit görünür ve korumayı zayıflatır. Kayıtları ele geçiren bir saldırgan aynı karmaları gruplayıp onları birlikte hedefleyebilir. Tek bir tahmin o zaman birçok hesabı açığa çıkarabilir. Bu zayıflıktan kaçınmak için her kayıt için her zaman farklı bir tuz değeri kullan.



Özetliyorsun: parolaları asla düz metin olarak saklama çünkü ele geçirilen kayıtlar okunabilir olurdu. Bunun yerine her kayıt için farklı bir tuzla karma işlemi yap, oturum açmada işlenmiş girdiyi karşılaştır ve basit hızlı karmaların tek başına yeterince korumayabileceğini unutma. Aksi halde tek bir sızıntı her kullanıcıyı aynı anda açığa çıkarırdı. Bu yolu izle, o zaman sistemin orijinal parolayı hiç geri okuması gerekmez.