Hashing

Was ist Passwort-Hashing?

Damit Passwörter in der Datenbank nicht im Klartext angezeigt werden müssen diese gehasht werden. Somit wird verhindert das der Hacker bei einem Hackerangriff alle Passwörter der bestehenden User sehen kann.

Was ist ein Hash?

Als **Hashing** bezeichnet man die Umwandlung einer Zeichenfolge in einen numerischen Wert oder Schlüssel mit fester Länge. Der numerische Wert ist der Hashwert und eine andere Darstellung der ursprünglichen Zeichenfolge.

Für welche Anwendungen wird der Hash benutzt?

- Datentyp Hashtabelle
- Caching
- Schutz sensibler Daten
- Auffinden von Duplikaten
- Suche nach ähnlichen Datensätzen oder Substrings in Zeichenketten

Wie funktioniert das Hashing?

Mittels eines Password Hashing-Algorithmus werden Passwörter in einer festgelegten Codefolge in zufälligen Buchstaben und Zahlen umgewandelt. Die Generierung der sogenannten Hashes läuft also automatisiert mit einem Hash-Algorithmus ab.

Bekanntesten Hash-Algorithmen

- Argon2
- Bycrypt
- Scrypt
- PBKDF2

Neben der Algorithmus kommt es beim Hashen noch auf andere Faktoren drauf an. Die Länge des zu hashenden Passwortes sollte mindestens 8 Zeichen lang sein, um Dictionary Angriffe zu erschweren.

Anwendung von Hashwerten

Als ein einfaches Beispiel für die Verwendung von Hash Werten kann eine Gruppe von Städten in einer Datenbank dienen, die wie folgt angeordnet ist:

Aachen, Bonn, Gelsenkirchen - und viele mehr in alphabetischer Reihenfolge.

Jeder dieser Namen ist der Schlüssel für die Daten, die zu dieser Stadt in der Datenbank gespeichert sind. Bei einer Suche nach einem bestimmten Eintrag müsste zunächst zeichenweise nach Übereinstimmungen im Namen gesucht werden. So lange, bis eine Übereinstimmung gefunden wird und die anderen Einträge ausgeschlossen werden konnten.

Mithilfe eines Hash-Algorithmus wird für jeden Namen ein **eindeutiger, mehrstelliger Schlüssel**, der Hashwert, generiert.

Zum Beispiel: 7864 Aachen, 9802 Bonn, 1990 Gelsenkirchen etc.

Hashwerte zum Verschlüsseln digitaler Signaturen

Verschlüsselungsprogramme nutzen Hashwerte zum Verschlüsseln und Entschlüsseln digitaler Signaturen - zum Beispiel zum Authentifizieren von Absendern und Empfängern von Nachrichten.

Die Hash-Funktion transformiert zunächst die digitale Signatur. Dann werden sowohl der berechnete Hash-Wert - der auch als Message Digest bezeichnet wird - als auch die Signatur in separaten Übertragungen an den Empfänger gesendet.

Entschlüsselung durch den Empfänger

Unter Verwendung der gleichen Hash-Funktion, die der Absender verwendet hat, leitet der Empfänger aus der Signatur einen Message Digest ab und vergleicht diesen mit dem Message Digest, den er ebenfalls empfangen hat.

Sie sollten beide übereinstimmen. Stimmen die beiden Werte nicht überein, ist dies ein Zeichen für eine Manipulation.

Anforderungen an eine Hashfunktion

Eine Hashfunktion muss deterministisch sein. Das bedeutet, dass die Funktion auch bei einer mehrfachen Anwendung für jede einzelne Eingabe dasselbe Ergebnis liefern muss.

Bei der Berechnung von Hashwerten dürfen keine zufälligen Elemente mit in die Berechnung einfließen, wie zum Beispiel bei einer Randomisierung Funktion. Es muss zudem sichergestellt sein, dass eine andere Hashfunktion den Hashwert nicht umkehren und damit entschlüsseln kann.

Passwort-Salt

Da ein Algorithmus gleiche Passwörter immer gleich Hashed und man somit leicht das Klartext Passwort raus filtern kann wird der Hash noch ”gesalzen”. Ein Salt ist eine zufällig generierte Ziffer, die in dem Hash einfließt. Somit kann also verhindert werden das der gleiche Hash öfter in der Datenbank steht.

Password- Pepper

Auch beim Peppered-Hash wird eine zufällige Zahlenfolge in den Hash mit eingeflossen. Jedoch anders als beim Salt, speichert die Datenbank den Pepper nicht zusammen mit den Login-Daten. Dieser wird sicher auf getrennt mit den Passwort auf einen anderen Ort bewahrt.

Unterschied Hashing Verschlüsseln

Der Grundlegende Unterschied zwischen Hashing und Verschlüsseln ist, das ein Hash im Gegensatz zu einer Verschlüsselung nicht rückgängig gemacht werden kann. Hashwerte müssen somit nicht geheim gehalten werden da kein Rücklauf zum Klartext möglich ist.