1. Heširanje lozinki

Heširanje je ključni korak u zaštiti poverljivih podataka, kao što su lozinke. Neke od najboljih praksiza heširanje lozinki uključuje korišćenje jakih, standardnih algoritama kao i implementaciju dodatnih mera kako bi se obezbedila njihova poverljivost. Danas, neki od najčešće korišćenih algoritama za heširanje jesu *Argon2id* i *bcrypt.* Aspekti kao što su bezbednost, performanse i lakoća korišćenja mogu uticati na izbor algoritma u zavisnosti od toga šta nam je prioritet. Iako *bcrypt* može biti preferiran zbog boljih performansi i šire podrške, kada je u pitanju bezbednost, *Argon2id* se izdvaja kao bolji izbor. *Argon2id*, iako je sporiji, namerno je dizajniran da bude memorijski i računarski intenzivan, što ga čini otpornijim na *brute force* napade.

Prilikom implementacije, važno je obratiti pažnju na konfiguraciju i adekvatan odabir parametara. Preporučena praksa uključuje podešavanje parametara poput minimalnog broja iteracija, minimalne veličine memorijskih zahteva i stepen paralelizma na odgovarajuće vrednosti koje odgovaraju specifičnim potrebama za sigurnošću. Neke od preporučenih vrednosti jesu:

* Memorija: 46 MiB, Broj iteracija: 1, Stepen paralelizma: 1
* Memorija: 19 MiB, Broj iteracija: 2, Stepen paralelizma: 1
* Memorija: 12 MiB, Broj iteracija: 3, Stepen paralelizma: 1
* Memorija: 9 MiB, Broj iteracija: 4, Stepen paralelizma: 1
* Memorija: 7 MiB, Broj iteracija: 5, Stepen paralelizma: 1

Prilikom implementacije *Argon2id* algoritma, preporučuje se korišćenje kriptografskih biblioteka i okvira, poput *libsodium*. Ova biblioteka se aktivno održava i ažurira, što osigurava primenu najnovijih bezbednosnih ispravki i unapređenja. Stoga, poslednja verzija *libsodium* za implementaciju *Argon2id* algoritma nema ozbiljnih ranjivosti i sigurna je za upotrebu.

Na osnovu gore navedenih stavki, sledeći zahtevi čine osnovu za bezbednu primenu mehanizma heširanja:

* Upotreba *Argon2id* algoritam za heširanje lozinke, koji pruža visok nivo bezbednosti.
* Konfiguracija parametara poput troškova vremena, troškova memorije i paralelizam na odgovarajući način kako bi se postigla ravnoteža između bezbednosti i performansi.
* Implementiacija *Argon2id* algoritmakoristeći pouzdan provajder kao što je *libsodium* ili druga dobro prihvaćena kriptografska biblioteka.
* Redovno praćenje bezbednosnih saveta i ažuriranja od provajdera kako bi se brzo rešili svi eventualni propusti.
* Osigurati da implementacija prati najbolje prakse za skladištenje lozinki, uključujući sigurno slanje i heširanje lozinki pre nego što se sačuvaju u bazi podataka.
* Primena praksi bezbednog kodiranja i sprovođenje redovnih bezbednosnih revizija radi identifikacije i adresiranja potencijalnih ranjivosti u implementaciji.

Pridržavajući se ovih zahteva, dizajniran mehanizam heširanja može efikasno zaštititi poverljivost korisničkih lozinki, istovremeno smanjujući potencijalne bezbednosne rizike.