Authenticatie: Analyse

Team Data Dumpsters:

Niek Smets

Lorenzo Elias

Nicolas Van Dyck

Matthias Van Rooy

Ward Boeckx

DATA DUMPSTERS

Inhoud

1. Inleiding

Omdat we met gevoelige gegevens werken is authenticatie een belangrijk onderdeel van onze applicatie. Gebruiksvriendelijk is ook een belangrijk aspect van de applicatie, daarom moeten we de juiste balans zoeken tussen veiligheid en gebruiksvriendelijkheid. Met dit in gedachten zijn wij tot de volgende oplossing gekomen. Een combinatie van authenticatie met gebruikersnaam + wachtwoord en een OTP (One-Time-Password). Deze oplossing biedt voldoende beveiliging zonder het gebruiksgemak zodanig te hinderen dat het aanmelden irritant wordt.

2. Veilig omgaan met gebruikersnamen en wachtwoorden

Bij het gebruik van een combinatie van gebruikersnaam en wachtwoord is het essentieel om deze gegevens op een veilige manier te verwerken en op te slaan. Binnen onze applicatie zorgen we hiervoor door:

- **Hashing en salting van wachtwoorden:**
  Wachtwoorden worden nooit in leesbare tekst opgeslagen. In plaats daarvan worden ze gehashed met een cryptografisch veilige hashfunctie (bijvoorbeeld bcrypt of Argon2). Om extra beveiliging te bieden tegen aanvallen, voegen we een unieke **salt** toe aan elk wachtwoord voordat het wordt gehashed. Dit betekent dat zelfs als twee gebruikers hetzelfde wachtwoord kiezen, de opgeslagen hash verschillend zal zijn. Bibliotheken zoals bcrypt gebruiken salting intern tijdens het hashen.
- **Beveiliging tegen brute-force aanvallen:**
  Door het gebruik van veilige hashing-algoritmes wordt de rekenkracht die nodig is om wachtwoorden te raden aanzienlijk verhoogd. Dit maakt brute-force aanvallen vrijwel ondoenlijk.
- **Validatie van wachtwoordsterkte:**
  Bij het aanmaken van een wachtwoord zorgen we dat gebruikers een sterk wachtwoord kiezen, met minimale lengte en voldoende variatie (cijfers, hoofdletters, symbolen).

Door deze technieken te combineren met een OTP, bieden we een robuuste beveiliging zonder de gebruikerservaring onnodig te compliceren.

3. Wat is een OTP?

Een **OTP** (One-Time Password) is een eenmalige code die wordt gebruikt om een gebruiker te verifiëren tijdens een beveiligingsproces. Het is meestal een willekeurig gegenereerde reeks cijfers of letters en is slechts korte tijd geldig. OTP's worden vaak gebruikt in situaties zoals:

- **Twee-factor-authenticatie (2FA):**

Je ontvangt een code via e-mail, sms, of een app (zoals Google Authenticator) om naast je wachtwoord in te loggen.

- **Verificatie van identiteit:**

Bijvoorbeeld tijdens het aanmaken van een account of wijzigen van een belangrijk gegeven, zoals een wachtwoord.

- **Beveiliging van transacties:**

Banken sturen vaak een OTP om betalingen of gevoelige acties te autoriseren.

In onze applicatie gaan wij een OTP gebruiken als 2FA.

4. Voordelen van OTP’s voor de gebruiker

Het gebruik van OTP’s biedt ook specifieke voordelen voor de eindgebruiker:

- **Gebruiksvriendelijkheid:** Het proces is eenvoudig en intuïtief. Gebruikers ontvangen een code op een vertrouwd kanaal zoals e-mail of sms en hoeven geen extra software of apparaten aan te schaffen.
- **Flexibiliteit:** OTP’s zijn platformonafhankelijk en kunnen gemakkelijk worden geïntegreerd in verschillende apparaten en applicaties. Hierdoor kan de gebruiker zich overal veilig aanmelden.
- **Beperkte impact op workflow:** De OTP-authenticatie voegt slechts een kleine extra stap toe, wat de impact op de gebruikerservaring minimaliseert.

5. Waarom een combinatie van gebruikersnaam + wachtwoord en OTP

De combinatie van een gebruikersnaam + wachtwoord met een OTP biedt een solide balans tussen gebruiksvriendelijkheid en beveiliging. Deze aanpak heeft de volgende voordelen:

- **Bescherming tegen wachtwoordlekken:** Mocht een wachtwoord uitlekken door phishing of een datalek, dan is alleen het wachtwoord niet voldoende om toegang te krijgen tot een account. De OTP fungeert als een tweede beveiligingslaag.
- **Extra beveiliging voor gevoelige gegevens:** Omdat onze applicatie met gevoelige gegevens werkt, is een extra beveiligingslaag noodzakelijk. OTP's verminderen het risico op ongeautoriseerde toegang aanzienlijk.
- **Eenvoudige implementatie:** OTP's kunnen eenvoudig worden gegenereerd en geleverd via bestaande kanalen zoals e-mail of sms. Dit maakt het toegankelijk zonder dat gebruikers extra hardware of geavanceerde technische kennis nodig hebben.

6. Beveiligingsstandaard in de industrie

De combinatie van gebruikersnaam + wachtwoord en OTP wordt gezien als een **best practice** binnen de industrie. Veel toonaangevende bedrijven en sectoren, zoals banken en gezondheidszorg, hanteren deze aanpak omdat het:

- **Aan compliance-eisen voldoet:** Voor bepaalde branches zijn beveiligingsnormen, zoals GDPR of ISO 27001, vereist. Het gebruik van 2FA via OTP kan helpen om aan deze normen te voldoen.
- **Weerbaarheid tegen veelvoorkomende aanvallen:** Deze methode beschermt tegen bedreigingen zoals brute-force aanvallen, credential stuffing en man-in-the-middle aanvallen.
- **Breed geaccepteerd is:** Gebruikers zijn vaak al bekend met OTP's vanuit andere applicaties, wat de acceptatie vergroot.

7. Referenties

ByteByteGo. (2024, 11 17). *Youtube | System Design: How to store passwords in the database?* Opgehaald van Youtube: https://www.youtube.com/watch?v=zt8Cocdy15c

SpringDeveloper. (2024, 11 11). *Youtube | SpringDeveloper*. Opgehaald van Youtube: https://www.youtube.com/watch?v=otdPVawMXUU&list=PLbevqUaC-YZwqCAAs9tkZZ4Kl3kkf4aXj&index=3

Tanzu, S. b. (2024, 11 11). *Spring Boot | Authentication | one time token*. Opgehaald van Spring Boot Docs: https://docs.spring.io/spring-security/reference/6.4/servlet/authentication/onetimetoken.html