Spring Security Working Flow

#Spring Security Flow - Step by Step

1. Client Request

- Client (browser, mobile app, etc.) ek HTTP request send karta hai, jaise /login ya kisi protected resource (/admin) ke liye.

2. Request Security Filter Chain mein Enter Karti Hai

- Request Security Filter Chain se guzarti hai, jisme multiple filters hote hain jo security checks perform karte hain.
- Sabse pehla filter SecurityContextPersistenceFilter hota hai, jo existing SecurityContext ko restore karta hai (session ya storage se).

3. Authentication Process Start Hoti Hai

- Agar request ko authentication ki zarurat hai, toh filter chain ka ek filter (e.g., UsernamePasswordAuthenticationFilter ya BearerTokenAuthenticationFilter) authentication request ko handle karta hai.
- Authentication ke liye AuthenticationManager ka method authenticate(Authentication authentication) call hota hai.

4. AuthenticationManager ke Under

- AuthenticationManager ek interface hai jiska commonly used implementation ProviderManager hota hai.
- ProviderManager ke paas ek list hoti hai AuthenticationProvider ki.
- Ye providers ko sequentially check karta hai ki kaunsa provider current authentication request ko handle kar sakta hai (supports() method ka use karke).

5. AuthenticationProvider

- AuthenticationProvider ek interface hai jo authentication ka actual kaam karta hai.
- Common implementations:
  - DaoAuthenticationProvider: UserDetailsService aur PasswordEncoder ka use karke user ko database ya in-memory store se verify karta hai.
  - Other providers: OAuth2AuthenticationProvider, JwtAuthenticationProvider, LdapAuthenticationProvider, etc.

6. Successful Authentication

- Agar koi AuthenticationProvideruser ko successfully authenticate kar deta hai:
  - Ek authenticated Authentication object create hota hai.

- Ye authenticated object SecurityContext me save hota hai (SecurityContextHolder ke through).

7. Forward to DispatcherServlet and Controllers

- Authenticated request DispatcherServlet tak forward ki jati hai.
- Controllers aur other request handlers request ko process karte hain.
- Agar user authorized hai, toh requested resource return hota hai; agar authorized nahi hai, toh "Access Denied" response milta hai.

8. Authorization Check

- Spring Security roles aur permissions ko validate karta hai (@PreAuthorize, @Secured, etc. ka use karke).
- FilterSecurityInterceptor final authorization check karta hai.

9. Response

- Agar authentication aur authorization successful hai, toh resource ya response user ko milta hai.
- Agar authentication fail hoti hai, toh appropriate error response diya jata hai (e.g., 401 Unauthorized ya 403 Forbidden).

Summary of Steps

1. **Client Request**: Client ek HTTP request send karta hai.
2. **Security Filter Chain**: Request security filters ke through pass hoti hai.
3. **Authentication Start**: Authentication ke liye specific filter kaam karta hai.
4. **AuthenticationManager**: AuthenticationManager AuthenticationProvider ko invoke karta hai.
5. **AuthenticationProvider**: User authentication ko verify karta hai.
6. **Successful Authentication**: SecurityContextHolder me authenticated object save hota hai.
7. **DispatcherServlet and Controllers**: Authenticated request controllers tak jati hai.
8. **Authorization Check**: User roles aur permissions check hote hain.
9. **Response**: Successful authentication ke baad resource ya error response return hota hai.

Ye pura process ensure karta hai ki sirf authorized users hi protected resources access kar sakein.

By Aron