**03.04.2023**

İlgili günde uygulama güvenliği için Spring Security paketi ile ilgili araştırma ve bu paketin ayar yapılandırması işlemi hakkında uygulama yapılmıştır. Geliştirilmekte olan Bulut Depolama uygulamasına entegrasyonu sağlanmıştır.

Spring Security; doğrulama (authentication), yetkilendirme (authorization), şifreleme(password encoder) ve CSRF gibi güvenlik önlemleri sağlayan ve bir uygulamanın güvenliği belli bir standarda oturtmayı sağlayan pakettir.

Sıragelen bir Web uygulamasında temel doğrulama ve yetkilendirmenin olmasına rağmen saldırganlar farklı saldırı yöntemleri kullanarak güvenlik açıkları ortaya çıkarmaktadır. Saldırganların sürekli olarak yenisini çıkardığı bu saldırıları proje geliştiricisi olarak takip etmek oldukça zordur. Spring Security’nin kullanılmasının bu konudaki yararı; paketin sürekli olarak uygulama çalıştıkça Maven üzerinden güncellenebilmesidir. Bu sayede yeni güvenlik açıkları otomatize bir şekilde kapatılabilir ve yeni güvenlik teknolojileri geliştirmiş olduğumuz Web uygulamalarına uygulama yapısını bozmadan entegre edilebilir.

Spring Security; Spring Initializr aracı ile Spring Boot projesine bağımlılık olarak eklenebilmektedir. Spring Security paketini projeye bağımlılık olarak ekledikten sonra gerekli yapılandırmalar yapılmalı ve aktif edilmelidir.

Spring Security’nin yapılandırılması için gerekli kodlara aşağıda yer verilmiştir. Proje içerisinde WebSecurityConfig isminde bir sınıf oluşturulmuştur. Bu sınıf WebSecurityConfigurer sınıfından kalıtım almaktadır. Sınıfın yapılandırma sınıfı olduğunu belirtmek için sınıf üstüne @Configuration anotasyonu eklenmiştir. Spring Security’nin proje içerisinde aktifleşmesi için @EnableWebSecurity anotasyonu da eklenmiştir. Kullanıcının kayıt olurken girmiş olduğu parolaların şifrelenmesi için BcryptPasswordEncoder sınıfından nesne döndüren bir metot tanımlanmıştır. Parolalar Bcrypt algoritmasına göre şifrelenerek veritabanında saklanacaktır. HTTP protokolü ile gelen isteklerin güvenliği ve yapılandırması için “protected void configure(HttpSecurity http)” metodu tanımlanmıştır. Bu metot içerisinde giriş sayfasının hangi URL adresine yapılacağı loginPage(“/login”) tanımı ile yapılmıştır. Kullanıcıdan giriş yaparken alınan kullanıcı adı ve parola bilgisinin hangi parametre ile geldigği **usernameParameter**() ve **passwordParameter**() tanımları ile belirlenmiştir. Bunun yanında giriş isteğinin hangi URL üzerinden geleceği de **loginProcessingUrl**() tanımı ile belirlenmiştir.

@Configuration

@EnableWebSecurity

public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

@Bean

public BCryptPasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder();

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception { auth.authenticationProvider(authenticationProvider());

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/users").authenticated()

.anyRequest().permitAll()

.and()

.formLogin()

.loginPage("/login")

.usernameParameter("username")

.passwordParameter("password")

.loginProcessingUrl("/process-login")

.defaultSuccessUrl("/home")

.permitAll()

.and()

.logout().logoutSuccessUrl("/").permitAll()

//API Authentication Enable

.and()

.cors().and().csrf().disable();

}

}