УДК004.622

А. В. Марущак
В. П. Майданюк

ФУНКЦІЇ БЕЗПЕКИ НА МОВІ JAVA

Вінницький національний технічний університет

Анотація

У роботі описано модуль безпеки персональних даних найбільш популярного фреймворку для розробки Java застосунків Spring. На момент написання актуальною є версія 5.6.2. У сучасному світі з розвитком технологій збільшуються випадки втрати особистої інформації, отже важливим аспектом у розробці програмних додатків є забезпечення користувацької безпеки.

Ключові слова: Spring 5.0, Spring Security, Java, захист даних, шифрування паролів, автентифікація, авторизація.

Abstract

The paper describes the personal data security module of the most popular framework for developing Java applications Spring. At the time of writing, version 5.6.2 is relevant. In today's world, with the development of technology, the number of cases of loss of personal information is increasing, so an important aspect in the development of software applications is to ensure user security.

Keywords: Spring 5.0, Spring Security, Java, data protection, password encryption, authentication, authorization.

Spring Security – це фреймворк, який надає функції безпеки, такі як: автентифікація, авторизація для створення програмних застосунків з використанням мови програмування Java. Авторизація – це процес, що дозволяє розробнику побудувати у запланованому програмному забезпеченні необхідну ієрархію користувачів з різним доступом до виконання дії. Автентифікація – це додатковий процес для успішного проходження авторизації, який повинен забезпечувати правильне розпізнання та ідентифікацію кожного потенційного користувача, який намагається отримати доступ до системи [1].

Фреймворк SpringSecurity підтримує широкий спектр моделей поведінки, наявна можливість інтеграції з популярними технологіями, такими як: HTTP Basic, LDAP, OpenID, AppFuse.

Перевагами у використанні даного модуля безпеки є повна підтримка автентифікації та авторизації користувачів, опрацювання даних в окремому потоці, інтеграція API Servlet, підтримка Spring MVC, портативність та мультиплатформність, повноцінна підтримка конфігурації Java.

Основний функціонал програмного доповнення Spring Security [1]:

- LDAP (полегшений протокол доступу до каталогів) – це відкритий прикладний протокол для підтримки та доступу до інформаційних служб розподілених каталогів через інтернет-мережу;
- єдиний вхід – ця функція дозволяє користувачеві отримати доступ до кількох програм за допомогою одного облікового запису, отже наявна можливість застосовувати єдиний логін та пароль для доступу до різних ресурсів;
- запам’ятовування користувача – реалізований даний функціонал за допомогою файлів cookie HTTP. Надає можливість системі запам’ятати визначеного користувача та уникати повторного введення персональних даних.
- OAuth 2.0 – ця функція надає користувачеві увійти в програму, використовуючи наявний обліковий запис соціальних мереж, GitHub, Google. Для правильного функціонування даної функції потрібно увімкнути двоступеневу автентифікацію за допомогою коду.

Починаючи з версії Spring Security 5.0, додана можливість забезпечити реактивне програмування та підтримку реактивного веб-виконання, також дана система може інтегруватися з Spring WebFlux.

У поточній версії Spring Security 5.0 було оголошено Password Encoder як застарілий [2]. Це був логічний крок, адже такий підхід не був оптимізований для випадково генерованого ключа шифрування. Отже, було змінено спосіб обробки закодованих паролів. У попередніх версіях кожна програма використовувала лише один алгоритм кодування пароля. За замовчуванням виконувалося це за допомогою Standard Password Encoder. Для кодування використовувався SHA-256 алгоритм. Для вбудування нового функціоналу використано концепцію делегування кодування пароля. Такий підхід дав можливість використовувати різні кодування для різних паролів. Spring розпізнає алгоритм за ідентифікатором із префікса закодованого пароля. Наприклад,

{bcrypt}$2b$12$FaLabMRystU4MLAasNOKb.HUElBAabuQdX59RWHq5X.9Ghm692NEi – пароль закодовано алгоритмом «bcrypt». На початку у фігурних дужках вказано тип використаного алгоритму під час шифрування, за допомогою цього ідентифікатора декодер розуміє як потрібно розшифровувати отриманий код.

Додавання конфігурації делегування паролів у програмне забезпечення є не важким процесом. Якщо хеш пароля не має префікса, процес делегування використовує алгоритм за замовчуванням. Отже, за замовчуванням буде використовуватися Standard Password Encoder. Таке рішення робить нові версії програмного забезпечення повністю сумісними із конфігурацією минулих версій. У версії Spring 5 представлено Password Encoder Facoryes. Create Delegating Password Encoder(). Даний вбудований метод повертає налаштований екземпляр класу Delegation Password Encoder [2]. Для паролів без префікса буде виконуватися поведінка за замовчуванням, а для хешів паролів, які містять префікс, делегування виконується відповідно передбаченого алгоритму. У Spring Security 5.0 додано такі методи шифрування [2]: bcrypt–Bcrypt PasswordE ncoder; ldap–Ldap Sha Password Encoder; MD4 - Md4 Password Encoder; MD5 – new Message Digest Password Encoder ("MD5"); noop–No Op Password Encoder; pbkdf2 - Pbkdf2 Password Encoder; scrypt–Scrypt Password Encoder; SHA-1 – new Message Digest Password Encoder ("SHA-1"); SHA-256 – new Message DigestPassword Encoder ("SHA-256"); sha256 –Standard Password Encoder; argon2 - Argon2 Password Encoder. Звичайно, було передбачено, що алгоритм виконання можна змінювати. Наприклад, є задача, де:

- Bcrypt – нове значення за замовчуванням;
- Scrypt – альтернатичний алгоритм;
- SHA-256 – поточний алгоритм.

Для такого випадку конфігураційний метод програмного застосунку буде мати структуру, як показано на рисунку 1.

```
@Bean
public PasswordEncoder delegatingPasswordEncoder() {
    PasswordEncoder defaultEncoder = new StandardPasswordEncoder();
    Map<String, PasswordEncoder> encoders = new HashMap<>();
    encoders.put("bcrypt", new BCryptPasswordEncoder());
    encoders.put("scrypt", new SCryptPasswordEncoder());

    DelegatingPasswordEncoder passworEncoder = new DelegatingPasswordEncoder(
      "bcrypt", encoders);
    passworEncoder.setDefaultPasswordEncoderForMatches(defaultEncoder);

    return passworEncoder;
}
```

Рисунок 1 – Конфігурація делегування паролів

Отже, розглянуто потужний фреймворк для побудови застосунків з використання мови програмування Java. Детально проаналізовано переваги та технології захисту персональних даних з використанням модуля Spring Security, розглянуто новий функціонал кодування паролів, який доступний у поточній версії Spring 5.6.2. Внесено зміни у стандартну конфігурацію програмного модуля Spring Security та отримано індивідуальний алгоритм обробки паролів, який надав змогу обробляти вхідні хеші паролів з урахуванням конфігурації програмного забезпечення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Spring Security. [Електронний ресурс]. – 2022 – Режим доступу до ресурсу: https://docs.spring.io/spring-security/reference/index.html.

2. What is Spring security. [Електронний ресурс]. – 2021 – Режим доступу до ресурсу: https://www.javadevjournal.com/spring/what-is-spring-security/.

Марущак Артем Володимирович - студент групи 3ПІ-19б, факультет інформаційних технологій та комп'ютерної інженерії**,** Вінницький національний технічний університет, Вінниця, e-mail: artem_marushchak@icloud.com

Майданюк Володимир Павлович - канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет.