УДК 004.031.2

С.Г. Московко

АВТОРИЗАЦІЯ ТА АВТЕНТИФІКАЦІЯ В ПРЕДМЕТНО-ОРІЄНТОВАНОМУ ПРОЕКТУВАННІ

Вінницький національний технічний університет

Анотація

В роботі проведено огляд реалізації автентифікації та авторизації в предметно-орієнтованому проектуванні, базуючись на піддомені ідентифікації та керування доступом. Досліджено такі моделі, як IBAC, RBAC, ABAC, PBAC і ZBAC, проведена оцінка переваг цих функцій в окремих обмежених контекстах.

Ключові слова: предметно-орієнтоване проектування, авторизація, автентифікація, управління ідентифікації та доступом.

Abstract

This article delves into authentication and authorization implementation in Domain-Driven Design, based on the Identity and Access Management subdomain. It explores models like IBAC, RBAC, ABAC, PBAC, and ZBAC, and emphasizes the benefits of segregating these functions within separate, reusable, and maintainable bounded contexts.

Keywords: domain-driven design, authorization, authentication, identity and access management.

Вступ

Предметно-орієнтоване проектування (DDD) — це підхід до розробки програмного забезпечення, який визначає пріоритет «логіки домену» або основних бізнес-концепцій створюваної системи. З точки зору безпеки, важливі два аспекти: авторизація та автентифікація. Авторизація визначає, які дії дозволено виконувати користувачеві в системі, а автентифікація перевіряє особу користувача. Ці функції життєво важливі в багатьох програмах, особливо в тих, що мають справу з конфіденційними даними користувача.

Метою даної роботи є дослідження та роз'яснення процесу впровадження аутентифікації та авторизації в рамках предметно-орієнтованого проектування. Робота спрямована на детальному аналіз моделей контролю доступу, таких як IBAC, RBAC, ABAC, PBAC та ZBAC.

Результати роботи

Керування ідентифікацією та доступом (IAM) є критично важливим субдоменом у DDD. Автентифікація та авторизація зазвичай обробляються в цьому контексті. Як піддомен компонент IAM має бути розроблений і реалізований окремо від решти системи, забезпечуючи її високу цілісність і слабку залежність.

Для реалізації авторизації та автентифікації можна використовувати декілька моделей, зокрема контроль доступу на основі ідентифікації (IBAC), контроль доступу на основі ролей (RBAC), контроль доступу на основі атрибутів (ABAC), контроль доступу на основі політики (PBAC) і нульовий контроль. Контроль доступу на основі довіри (ZBAC). Вибір моделі залежить від системних вимог.

IBAC (Identity-Based Access Control) — У цьому типі системи контролю доступу, доступ надається або відмовляється на основі ідентичності користувача. Це найпростіший метод контролю доступу, в якому кожен користувач отримує унікальну ідентифікацію (наприклад, ім'я користувача та пароль), яка використовується для перевірки того, чи має він право доступу до певного ресурсу. Цей вид системи є простим у розумінні та впровадженні, а також забезпечує ясність, оскільки кожен користувач має унікальні облікові дані. Проте, IBAC може бути важким для управління в великих системах і не надає гнучкості, оскільки доступ базується лише на ідентичності користувача.

RBAC (Role-Based Access Control) - це система контролю доступу, в якій дозволи на доступ до системи визначаються на основі ролей користувачів у системі. Користувачі отримують роль, яка в свою чергу пов'язана з певними дозволами. Це дозволяє легко керувати групами користувачів, які мають схожі дозволи доступу. RBAC легше масштабувати, ніж IBAC, і забезпечує ясність у відслідковуванні доступу. Однак, воно може бути негнучким, якщо користувачам потрібен доступ, який виходить за рамки їхньої ролі, і може створити проблеми при управлінні великою кількістю ролей.

ABAC (Attribute-Based Access Control) - це більш гнучка та гранульна система контролю доступу, ніж RBAC. Вона визначає доступ на основі атрибутів користувача, ресурсу, дії та, можливо, іншого контексту. Політики доступу в ABAC можуть бути висловлені як правила, що дозволяють чи забороняють доступ на основі будь-якої комбінації атрибутів. Але системи ABAC можуть бути вкрай складними для налаштування та управління, і додаткова обробка, необхідна для перевірки атрибутів та правил, може сповільнити систему.

PBAC (Policy-Based Access Control) - це метод контролю доступу, який використовує бізнес-правила для визначення того, чи дозволяється доступ. Правила можуть бути висловлені в термінах атрибутів користувача, ресурсу, дії і контексту, подібно до ABAC, але вони також можуть включати бізнес-логіку, таку як схвалення рівнів або виключення часу. Однак, створення та управління бізнес-правилами може бути складним, і управління великим набором бізнес-правил може бути трудомістким та часовитратним.

ZBAC (Zero Trust Access Control) - це стратегія безпеки, що ґрунтується на принципі "нульової довіри", який передбачає не довіряти нічому за межами або всередині мережі за замовчуванням. Замість надії на захист мережевого периметра, ZBAC вимагає постійної аутентифікації та перевірки, незалежно від місця знаходження або статусу користувача. ZBAC може знизити ризик вторгнень і дає кращий контроль над доступом до конкретних ресурсів. Проте, ZBAC може бути складним для імплементації та вимагає високого рівня управління, а постійна перевірка аутентифікації може бути незручною для користувачів.

Загалом в контексті предметно-орієнтованого проектування, керування ідентифікацією та доступом є загальним субдоменом. Це означає, що однакові принципи та практики дизайну застосовуються до багатьох різних доменів, незважаючи на їх специфіку.

Реалізація IAM як окремого обмеженого контексту дозволяє зосередитися на цих універсальних принципах, ефективно абстрагуючись від особливостей домену. Це може бути значною перевагою з точки зору безпеки, багаторазового використання коду та зручності обслуговування.

Наприклад, можна було б інкапсулювати ці моделі авторизації в обмежений контекст IAM і надати уніфікований API для решти системи. Внутрішню реалізацію можна було змінити з однієї моделі на іншу, не впливаючи на інші частини системи.

Висновки

Предметно-орієнтоване проектування забезпечує чіткий підхід до реалізації автентифікації та авторизації та заохочує використання шаблонів, таких як обмежені контексти та піддомени, щоб ваша система була модульною, безпечною та зручною для обслуговування. Вибір IBAC, RBAC, ABAC, PBAC або ZBAC залежить від специфіки вашої системи, але підхід DDD гарантує, що ви можете міняти їх місцями за потреби, не порушуючи решту кодової бази.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Vernon V. Implementing Domain-Driven Design : Addison-Wesley Professional, 2013. 656 c. ISBN 978-0321834577.

2. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software, 2003. 320 c. ISBN 9780321125217.

3. Types of Access Control Systems. – URL: https://umbrellatech.co/access-control-systems-chicago-il/system-types/

4. Identity-based policies and resource-based policies. – URL: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html
5. Authorization and authentication in clean architecture – URL: https://lessthan12ms.com/authorization-and-authentication-in-clean-architecture.html

Московко Сергій Геннадійович — факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: cakedispensers@gmail.com.

Moskovko Serhii G. — Department of intelligent information technologies and automation, Vinnytsia National Technical University, Vinnytsia, e-mail: cakedispensers@gmail.com.