УДК 004.056.53:004.415.5

І. Р. Арсенюк
А. П. Стецюра

АРХІТЕКТУРА ПРОГРАМНОГО МОДУЛЯ ГІБРИДНОГО КОНТРОЛЮ ДОСТУПУ ДО GIT-РЕПОЗИТОРІЇВ

Вінницький національний технічний університет

Анотація

У роботі представлено результати проєктування програмного модуля гібридного контролю доступу до Git-репозиторіїв. Обґрунтовано доцільність розширення класичної рольової моделі атрибутивними політиками на рівні окремих гілок. Наведено архітектуру, що складається з клієнтського WEB-додатку, серверного API та автономного шлюзу доступу.

Інтеграція механізмів серверного перехоплення транзакцій дозволяє динамічно валідувати права користувачів з урахуванням часових вікон та масок гілок. Результати дослідження підтверджують ефективність застосування пріоритезації заборонних правил для забезпечення цілісності кодової бази. Отримані архітектурні рішення формують основу для створення безпечних корпоративних платформ хостингу коду.

Ключові слова: системи контролю версій, гібридний доступ, серверні перехоплювачі, шлюз доступу, мікросервісна архітектура, атрибутивна модель авторизації.

Abstract

This paper presents the results of designing a software module for hybrid access control to Git repositories. It substantiates the expediency of extending the classic role-based model with attribute-based policies at the level of individual branches. The presented architecture consists of a client web application, a server API, and an autonomous access gateway.

The integration of server-side transaction interception mechanisms enables the dynamic validation of user permissions, taking into account time windows and branch masks. The research results confirm the effectiveness of prioritizing deny rules to ensure the integrity of the codebase. The obtained architectural solutions form the foundation for developing secure corporate code hosting platforms.

Keywords: version control systems, hybrid access, server interceptors, access gateway, microservice architecture, attribute-based authorization model.

Вступ

Системи контролю версій є фундаментальним інструментом сучасного процесу розробки програмного забезпечення. Розподілена архітектура системи Git забезпечує високу стійкість до збоїв та зручність командної роботи шляхом повного копіювання репозиторію на локальні машини клієнтів. Однак базові механізми авторизації Git обмежуються бінарною моделлю, яка надає права на модифікацію виключно до всього сховища загалом [1]. В умовах корпоративної розробки виникає критична необхідність у забезпеченні посиленого контролю доступу безпосередньо на рівні окремих гілок.

Класична рольова модель управління доступом успішно вирішує завдання базової ідентифікації повноважень. Разом з тим вона не дозволяє гнучко враховувати динамічні атрибути середовища під час виконання транзакцій [2]. Стандартні інструменти не здатні автоматично обмежувати час виконання push операцій або надавати одноразові перепустки для термінового виправлення програмних помилок. Оскільки архітектура ядра Git не містить вбудованих засобів для такого рівня авторизації, розв'язання цієї проблеми делегується зовнішнім програмним комплексам [3].

З огляду на це виникає потреба у розробці гібридних систем контролю, які органічно поєднують надійність рольової моделі з гнучкістю атрибутивного доступу. В рамках цього дослідження пропонується спроєктувати автономний програмний модуль доступу, здатний виконувати мережеві транзакції та багатокритеріальну перевірку політик безпеки ще до змін у базі даних сервера.

Результати дослідження

Система гібридного контролю доступу базується на трьох програмних модулях. Клієнтський WEB-додаток відповідає за управління станами та безпечну взаємодію з API на основі алгоритмів дедуплікації мережевих запитів. Серверний програмний інтерфейс реалізує бізнес-логіку та взаємодіє з реляційною базою даних PostgreSQL для зберігання правил доступу [4]. Центральним вузлом виступає автономний шлюз доступу, який безпосередньо обробляє SSH запити та делегує валідацію транзакцій спеціальним механізмам серверних перехоплювачів [3]. Логіку мережевої взаємодії клієнта та сервера під час виконання операції запису та момент ініціалізації перехоплювача наведено на рис. 1.

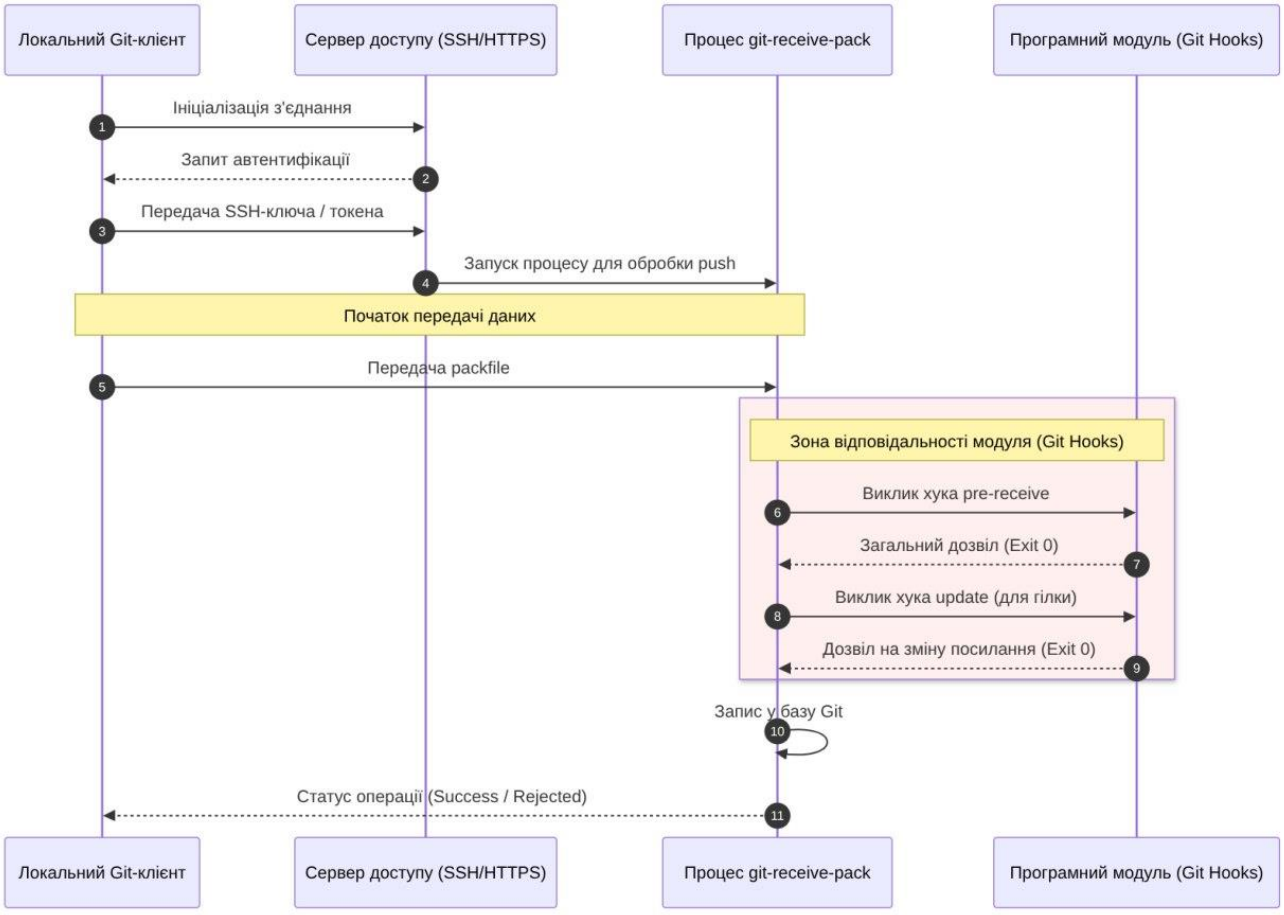

Рисунок 1 – Схема мережевої взаємодії клієнта та сервера під час виконання Git

Процес перевірки гібридних політик використовує комплексну оцінку прав доступу. Результат авторизації формується на основі перевірки множини ролей користувача, назви цільової гілки та інших атрибутів сесії. Головним принципом розв'язання логічних конфліктів між політиками дозволу та заборони є застосування принципу строгого домінування заборони. Це означає, що доступ до виконання транзакції надається виключно за умови наявності хоча б одного активного дозволяючого правила та повної відсутності будь-яких забороняючих правил.

Спроектована схема бази даних зберігає правила доступу в єдиній таблиці [4, 5]. Застосування жорстких перевірок на рівні ядра бази даних гарантує неможливість збереження хибних записів. Для підтвердження ефективності запропонованого гібридного підходу було проведено порівняльний аналіз із рольовою моделлю. Основні результати архітектурного порівняння наведено у таблиці 1.

Таблиця 1. Порівняльний аналіз моделей управління доступом у системах контролю версій

Критерій порівняння	**Рольова модель**	**Гібридна модель**
Гранулярність контролю	Рівень усього репозиторію	Рівень окремих гілок та їх масок
Облік контексту часу	Не підтримується	Динамічні часові вікна дії правил
Гнучкість винятків	Створення надлишкових ролей	Одноразові перепустки
Архітектурна складність	Низька	Висока
Пріоритезація конфліктів	Статична за ієрархією	Динамічна з домінуванням заборони

Аналіз даних підтверджує, що гібридна система забезпечує значно вищий рівень безпеки та гнучкості порівняно з класичними підходами, компенсуючи підвищену архітектурну складність розширеними можливостями адміністрування. Запропонована архітектура вирішує проблему надлишкових прав доступу та надійно захищає критичний код від небажаних змін. На відміну від існуючих рішень, система блокує порушення безпеки ще на рівні шлюзу, не потребуючи втручання у ядро Git.

Висновки

Проведене дослідження підтверджує, що розширення рольової моделі атрибутивними політиками суттєво підвищує безпеку систем контролю версій. Запропонована архітектура, спільно з алгоритмом домінування заборони, забезпечує автономну валідацію транзакцій та надійний захист кодової бази від несанкціонованих модифікацій без втручання у внутрішнє ядро Git. Отримані результати формують міцне інженерне підґрунтя для подальшого впровадження платформи у корпоративному середовищі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Pro Git Book. Git на сервері – Протоколи [Електронний ресурс] – Режим доступу: https://git-scm.com/book/uk/v2/Git-на-сервері-Протоколи
2. Pro Git Book. Customizing Git - Git Hooks [Електронний ресурс] – Режим доступу: https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks
3. Вікіпедія. Secure Shell (SSH) [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki /Secure_Shell
4. Вікіпедія. PostgreSQL [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/ PostgreSQL
5. Amazon Web Services. Role-Based Access Control (RBAC) vs. Attribute-Based Access Control (ABAC) [Електронний ресурс] – Режим доступу: https://aws.amazon.com/identity/access-management/rbac-vs-abac/

Арсенюк Ігор Ростиславович – канд. техн. наук, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, e-mail: air@vntu.edu.ua

Стецюра Антон Павлович – студент групи 3КН-22б, факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: anton.stetsyura.pr1@gmail.com

Arseniuk Igor R. – Associate Professor of the Computer Sciences Department, Vinnytsia National Technical University, Vinnytsia, e-mail: air@vntu.edu.ua

Stetsiura Anton P. – Student of the 3CS-22b group, Department of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: anton.stetsyura.pr1@gmail.com