3. Weber, M., et al. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks. ACM KDD Workshop on Deep Learning Day.

4. Tao, Bishenghui, Ivan Wang-Hei Ho, and Hong-Ning Dai. Complex network analysis of the bitcoin blockchain network. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2021. p. 1-5. https://ira.lib.polyu.edu.hk/bitstream/10397/107099/1/Ho_Complex_Network_Analysis.pdf.

5. Drobyk, O. V., Hashko, A. O. (2025). Integration of hybrid mathematical models for cluster-based risk detection in multi-network blockchain environments. In: Proceedings of the 1st International Scientific and Practical Conference "Applied Control Systems and Robotics", Kyiv, Ukraine, (page# 185).

6. Bondarchuk, A., Hashko, A., et al. (2025). Security challenges of blockchain interoperability mechanisms. In: Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II). CEUR Workshop Proceedings, Vol. 4145. URL: https://ceur-ws.org/Vol-4145/paper17.pdf.

7. Гашко А. О., Стражніков А. А. (2025). Порівняння алгоритмів побудови кластерної моделі на базі набору даних (dataset), отриманого з bigdata. Зв'язок, (1), 49-54. URL: https: // con.dut.edu.ua / index.php / communication/article/view/2839/2736.

8. Liu, Chenlei, Yuhua Xu, and Zhixin Sun. "Directed dynamic attribute graph anomaly detection based on evolved graph attention for blockchain." Knowledge and Information Systems 66.2 (2024): 989-1010. https://www.researchsquare.com/article/rs-3212327/latest.pdf.

9. Hashko, A. O., & Bondarchuk, A. P. (2025). AUTOMATED METHOD FOR VERIFYING THE CORRECTNESS OF THE EXECUTION OF SMART CONTRACTS IN THE BLOCKCHAIN NETWORK. Сучасний захист інформації, (4), 37-43. DOI: 10.31673/2409-7292.2025.041204.

10. Zhang, M., Zhang, X., Zhang, Y., & Lin, Z. (2024, September). Security of cross-chain bridges: Attack surfaces, defenses, and open problems. In Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (pp. 298-316). https://dl.acm.org/doi/pdf/10.1145/3678890.3678894.

11. Hashko, A. O. (2025). The GRX model as a universal approach to multidimensional clustering and risk assessment in AML systems for crypto-assets. In: Proceedings of the 1st International Scientific and Practical Conference "Applied Control Systems and Robotics", Kyiv, Ukraine, (page# 204).

12. Kou, G., et al. (2021). Evaluation of machine learning methods for financial fraud detection under class imbalance. Expert Systems with Applications, 180, 115067. DOI: 10.1016/j.eswa.2021.115067.

13. Гашко, А. О., et al (2025). Автоматизований метод перевірки правильності виконання смарт-контрактів в блокчейн мережі. Телекомунікаційні та інформаційні технології, (1), 13-20. DOI: 10.31673/2412-4338.2025.014506.

14. Shantyr, Anton, et al. "Prediction of quality software quality indicators with applied modifications of integrated gradiates methods." Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 15.2 (2025): 139-146. http://doi.org/10.35784/iapgos.6892.

15. Чичкарьов, Євген, et al. "Метод вибору ознак для системи виявлення вторгнень з використанням ансамблевого підходу та нечіткої логіки." Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка» 1.21 (2023): 234-251. https://csecurity.kubg.edu.ua/index.php/journal/article/download/523/409.

16. Zhebka, Viktoriia, et al. "Methods for Predicting Failures in a Smart Home." Digital Economy Concepts and Technologies Workshop 2024. Vol. 3665. Germany, 2024. https://elibrary.kubg.edu.ua/id/eprint/48728/.

17. Ribeiro, M. T., Singh, S., Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the ACM SIGKDD Conference. https://doi.org/10.1145/2939672.2939778.

Надійшла до редакції (Received): 13.01.2026
Прийнята до друку (Accepted): 17.03.2026
Опубліковано онлайн (Available online): 30.03.2026

УДК 004.056.5:004.7:004.451
DOI: 10.31673/2409-7292.2026.010317

Дарієнко Д.Г., Когут Н.М.

РОЗРОБКА МЕТОДУ ВИБОРУ IDP ПРОВАЙДЕРА ДЛЯ ІНТЕГРАЦІЇ З DOCKER

У роботі досліджено проблему забезпечення безпеки процесу збірки контейнерів у середовищі Docker шляхом вибору оптимального провайдера ідентичності (IdP). Основна мета полягала у формуванні методу вибору IdP, здатного запобігати виконанню несанкціонованих операцій під час збірки та розгортання

контейнерів. Для ідентифікації ключових ризиків застосовано підхід Security Threat Oriented Requirements Engineering (STORE), що дало змогу виявити загрози, зумовлені помилками розробників, використанням ненадійних або скомпрометованих build-агентів, а також можливістю несанкціонованого ініціювання збірки. На основі аналізу цих загроз сформульовано вимоги до IdP: підтримка багатофакторної автентифікації, використання короткочасних та відкличних токенів, гнучкі моделі контролю доступу, інтеграція з CI/CD процесами та Kubernetes, а також централізований аудит і мапінг claims на політики авторизації. У результаті проведеного дослідження сформовано метод вибору постачальника системи керування ідентичністю (IdP) для інтеграції в середовище Docker builder з метою запобігання виконанню несанкціонованих операцій. На основі аналізу загроз, визначених у межах підходу Security Threat Oriented Requirements Engineering (STORE), було побудовано систему критеріїв і ваг, що відображають вплив кожного аспекту IdP на усунення ризиків, пов'язаних із неконтрольованими діями розробників, агентів збірки та компрометацією секретів. Порівняльний аналіз провайдерів ідентичності визначив найбільш відповідних провайдерів IDP, які забезпечують необхідну гнучкість і рівень безпеки.

Ключові слова: Single Sign-On, Docker, DevOps-безпека, STORE, контроль доступу, захист інформації.

Вступ

Розвиток контейнерних технологій і систем CI/CD створив потребу у посиленому контролі за процесом збірки програмного забезпечення. Docker, який є основою багатьох DevOps-конвеєрів, не має вбудованого механізму для розмежування прав користувачів та агентів збірки. Це призводить до того, що будь-який користувач із доступом до середовища може виконувати операції, які виходять за межі його відповідальності, що підвищує ризик несанкціонованих змін у коді, витоку секретів або компрометації процесу розгортання. Проблема набуває особливої актуальності в умовах розподілених команд розробки, де відсутність централізованої системи керування доступом та авторизації ускладнює контроль доступу до критичних елементів CI/CD-інфраструктури.

Постановка задачі

Метою дослідження є розроблення набору правил для вибору системи керування доступом (IdP), яку можна інтегрувати з Docker builder для реалізації RBAC моделі доступу. Для досягнення поставленої мети визначено низку завдань, що становлять логічну послідовність формування методу. Необхідно здійснити аналіз сучасних методів збору вимог до програмного забезпечення та обрати підхід, який найбільш придатний для інтеграції з Docker builder; дослідити потенційні загроз, характерні для процесу збірки контейнерів; сформувати метод підрахунку та оцінювання рішень, який дозволяє кількісно порівняти різні системи керування доступом за визначеними критеріями безпеки, інтегрованості, керованості, надійності та вартості; проаналізувати популярні засоби керування доступом та визначити найбільш придатний варіант для впровадження у середовище Docker builder.

Аналіз публікацій

Розмежування доступу в інженерних командах неможливе без чіткого визначення ролей і дозволів. Модель рольового керування доступом (RBAC) визначає механізм, у якому права доступу призначаються не окремим користувачам, а ролям, що відповідають їхнім функціям у системі. Користувач отримує дозвіл виконувати лише ті операції, які пов'язані з його роллю, що дає змогу централізовано й узгоджено керувати політиками безпеки.

Такий підхід спрощує адміністрування, знижує ризик помилкових налаштувань і підвищує масштабованість у великих організаціях. RBAC формалізує принцип найменших привілеїв, оскільки кожен користувач має лише мінімальний набір дозволів, необхідний для виконання своїх завдань. Це зменшує ймовірність зловживань і випадкових порушень безпеки, адже будь-які операції поза межами ролі заборонені. Ієрархічна структура ролей дозволяє створювати узагальнені рівні доступу, що спрощує контроль у складних системах і забезпечує відтворюваність політик [1].

Централізований провайдер ідентичності знімає з команди розробки обов'язок створювати та підтримувати власний контур автентифікації й керування життєвим циклом облікових записів. Централізація ідентичності та делегування автентифікації спеціалізованому IdP покращують керованість, узгодженість політик і відповідність

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

стандартам, зменшуючи операційні витрати та складність інтеграцій. Для підприємств це означає скорочення технічного боргу й швидше впровадження практик безпеки [2].

IdP також дає можливість використання єдиного входу. SSO – механізм, за якого користувач один раз автентифікується і далі отримує доступ до пов'язаних сервісів, не вводячи облікові дані повторно; сучасні реалізації спираються на стандарти на кшталт OpenID. SSO формує спільну точку контролю, де впроваджуються політики, багатофакторна автентифікація та моніторинг подій, а отже підсилює цілісність процесу доступу [3].

Єдиний вхід позитивно впливає на безпеку середовища розробки. Зменшення кількості автентифікацій знижує ризики фішингу та повторного використання паролів, а централізоване застосування MFA і політик можна поширювати на всі інструменти процесу створення програмного забезпечення. Важливо обмежувати тривалість сесій, запобігати повторному використанню облікових даних і забезпечувати ізоляцію між сервісами, що користуються спільною системою входу. Поєднання цих технічних заходів із багатофакторною автентифікацією дозволяє поєднати зручність централізованого доступу з високим рівнем безпеки та зменшити ризики атак, пов'язаних із викраденням облікових даних або підробкою сесій [4].

Docker як інструмент контейнеризації не має вбудованої технології обмеження доступу до власного API. Модель авторизації є грубою і потребує підключення зовнішніх плагінів або політик, щоб відокремити дозволені та заборонені команди для різних користувачів. Відсутність тонкого керування операціями і широка довіра до демонів підвищують ризики, тож інтеграція з зовнішнім IdP і політиками є критичною для безпечної експлуатації [5].

Поєднання RBAC для мінімізації привілеїв, централізованого IdP для уніфікованих політик та SSO для узгодженого контролю автентифікації створює цілісний фундамент безпеки. У середовищах, де розробники працюють із реєстрами контейнерів, процесами збірки та кластерними ресурсами, така архітектура дозволяє зменшити кількість потенційних точок компрометації, забезпечує прозорість і повторюваність надання доступу та усуває обмеження вбудованих механізмів Docker.

Збір вимог

Для формування методу вибору системи керування доступом, що інтегрується з Docker builder, необхідно визначити ефективні підходи до збору вимог. У цьому контексті доцільно розглянути шість найвідоміших методів, які застосовуються в інженерії програмного забезпечення: опитування користувачів, анкетування, спостереження, моделювання цілей, формування вимог безпеки, групові сесії.

Перший метод – це інтерв'ю із зацікавленими сторонами-розробниками, DevOps, адміністраторами безпеки, менеджерами. Під час інтерв'ю можна з'ясовувати, які операції у Docker вважають ризиковими, які права мають користувачі зараз, які обмеження хочуть бачити. Профільований аналіз статей показує, що інтерв'ю є найпоширенішим методом у вимірюванні вимог: вони дозволяють глибинно з'ясувати приховані припущення та конфлікти між цілями різних груп [6].

Другий метод – анкетування, де більша кількість учасників дає відповіді на стандартизовані запитання. Це корисно для охоплення широкого кола користувачів або груп, особливо якщо команда розподілена. Такий підхід дозволяє отримати дані (наприклад, оцінки важливості певних функцій або ризиків), також стандартизувати відповіді, що полегшує їх кількісний аналіз і виявлення закономірностей у потребах користувачів [7].

Третій – спостереження за виконанням завдань, також відомий як етнографічний підхід. Він використовується для виявлення реальних потреб користувачів шляхом безпосереднього аналізу їхньої поведінки у робочому середовищі. На відміну від анкетування чи інтерв'ювання, цей метод не покладається на суб'єктивні описи дій, а дозволяє дослідникові спостерігати, як користувачі фактично взаємодіють із системою, інструментами чи процесами. Завдяки цьому можна виявити латентні потреби – ті, про які користувачі не завжди можуть

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

прямо повідомити, але які проявляються у щоденній роботі. У межах інженерії вимог метод «task observation» допомагає визначити критичні точки у процесах, де виникають помилки, неефективні дії або обхідні рішення, що свідчить про недоліки поточної системи [8].

Четвертий – моделювання цілей, наприклад, KAOS, де через аналіз цілей системи та конфліктів між ними формуються вимоги. Метод дозволяє вивести властивості, які IdP має підтримувати, виходячи з цілей безпеки, контролю доступу і гнучкості [9].

П’ятий підхід – формування вимог безпеки за допомогою threat-oriented підходів, наприклад, методологія STORE (Security Threat Oriented Requirements Engineering). У цьому підході спочатку ідентифікуються потенційні загрози (наприклад, несанкціонована команда docker rm, перегляд образів, зміна тегів), для кожної загрози формуються вимоги захисту, які потім трансформують у вимоги до IdP (наприклад, мінімальні права, відгук сесій) [10].

Шостий – групові сесії. Це групові обговорення за участю ключових фахівців, де проводяться сесії мозковий штурм функцій, обмежень, сценаріїв використання, що допомагає узгодити бачення та виявити приховані вимоги [11].

Кожен із цих методів має свої переваги й обмеження: інтерв’ю глибоке, але не масове; анкети охоплюють далі, але менш глибоко; спостереження реальне, але затратне за часом; моделювання цілей – абстрактне, потребує досвідчених аналітиків; threat-орієнтований підхід – структурований для безпеки, але може пропустити побутові нюанси; групові сесії допомагають консолідувати позиції, але можуть домінувати сильні голоси.

Основна частина

Обґрунтування вибору методології збору вимог

Серед підходів до збору вимог під час розробки методології вибору системи керування доступом для дослідження обрано метод, орієнтований на виявлення вимог безпеки, зокрема методологія Security Threat Oriented Requirements Engineering (STORE). Причина такого вибору пов’язана з метою проєкту: провайдер ідентичності впроваджується для реалізації системи контролю доступу до функцій Docker у процесі збірки та розгортання контейнерів щоб покращити безпекові характеристики середовища. Основним пріоритетом є захист від несанкціонованих операцій – запуску збірки, зміни образів, використання невірних або скомпрометованих build-агентів.

Традиційні методи збору вимог, такі як інтерв’ю, анкетування чи спостереження, спрямовані переважно на виявлення бізнес-потреб і зручності користувачів. Вони менш ефективні, коли необхідно зосередитися на технічних ризиках, сценаріях атак та політиках найменших привілеїв, які мають ключове значення для захисту середовища Docker. Методологія STORE дозволяє почати роботу з детального аналізу потенційних загроз і шляхів їх реалізації. Такий підхід дає змогу визначити, які саме механізми автентифікації, авторизації, аудиту й управління токенами повинен підтримувати обраний IdP, щоб ефективно протидіяти конкретним ризикам, наприклад, несанкціонованому запуску контейнерної збірки або впровадженню шкідливого коду під час побудови образів. Виявлені загрози стають безпосередньою основою для формування вимог до провайдера, що гарантує їхню релевантність та повну відповідність завданню контролю доступу.

Отже, обрання threat-oriented методології, а саме STORE, є логічним і виправданим кроком. Вона забезпечує зосередженість на безпеці, точне визначення критичних функцій і дозволяє сформулювати вимоги до IdP, спрямовані винятково на впровадження системи контролю доступу до функцій Docker, що є ключовою метою дослідження.

Опис методу вибору IDP провайдера

Логіка оцінювання провайдерів ідентичності має будуватися як процес, який поєднує етапи фільтрації, зваженого оцінювання, практичної перевірки та врахування ризиків. Насамперед необхідно визначити мінімальні вимоги, без яких жоден провайдер не може бути розглянутий. До таких вимог належать підтримка сучасних протоколів автентифікації: OIDC, OAuth 2.0, SAML 2.0; наявність багатофакторної автентифікації; механізмів короткочасних

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

токенів з відкликанням, централізованого аудиту, а також інтеграції з CI/CD-інструментами через API або SCIM. Якщо провайдер не відповідає хоча б одній із цих умов, він автоматично відхиляється на першому етапі.

Після попереднього відбору застосовується основна оціночна модель, що ґрунтується на зваженому скорингу. Для кожного провайдера формується набір критеріїв, яким присвоюються ваги залежно від їхньої важливості. Найвагомішими є критерії безпеки, інтеграції з Docker builder та CI/CD, операційної керованості, надійності та вартості володіння. Безпека має найбільший вплив на підсумкову оцінку, адже визначає здатність IdP запобігати несанкціонованим збіркам, контролювати дії агентів і гарантувати коректність політик доступу. Інтеграційні можливості мають другорядну, але суттєву вагу, оскільки саме вони визначають, наскільки просто впровадити IdP у вже існуючу DevSecOps-інфраструктуру. Менші, але не другорядні коефіцієнти присвоюються критеріям керованості, надійності та вартості, які впливають на довгострокову експлуатацію системи.

Розрахунок загальної оцінки для кожного провайдера здійснюється за формулою:

$$S = \sum_{i=0}^{n} (w_i \times s_i),$$

де n – кількість критеріїв оцінювання -1, w_i – вага і-го критерію, а s_i – нормована оцінка (від 0 до 1), отримана в результаті аналізу конкретного параметра. Цей підхід дозволяє об’єктивно порівняти різні рішення, враховуючи не лише функціональність, але й відповідність практичним вимогам середовища збірки контейнерів.

Аналіз загроз безпеці у процесі збірки контейнерів

Представлена таблиця узагальнює ключові загрози безпеці, що виникають у процесі збірки контейнерів у середовищі Docker. Її формування базується на результатах аналізу архітектури Docker builder та обмежень його вбудованої системи контролю доступу. Основна передумова полягає в тому, що Docker не має нативного механізму обмеження переліку операцій, які може виконувати користувач або агент під час збірки. Це створює можливості для зловживань, ескалації привілеїв, витоку даних і виконання несанкціонованих дій у межах конвеєра CI/CD.

Таблиця 1

Ключові загрози у середовищі Docker

Об’єкт	**Загроза**	**Суб’єкт**	**Наслідок**	**Ймовірність**	**Вплив**
Docker builder	Несанкціонований запуск збірки	Зовнішній зловмисник або помилка розробника	Вбудовування шкідливого коду в образ	Висока	Критичний
CI/CD агент	Використання неперевіреного агента збірки	Зовнішній зловмисник або помилка розробника	Підміна процесу збірки	Середня	Високий
Dockerfile	Помилки конфігурації (USER root, зайві порти)	Розробник	Ескалація привілеїв, компрометація хоста	Висока	Середній
Секрети збірки	Витік облікових даних через ARG/ENV	Розробник	Розкриття конфіденційних даних	Середня	Високий
Журнали збірки	Витік даних у логах	Агент або адміністратор	Компрометація секретів	Середня	Середній

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

У таблиці кожна загроза розглядається як комбінація об’єкта, суб’єкта, наслідку, ймовірності та впливу на систему. Об’єктом виступає компонент середовища збірки (Docker builder, CI/CD агент, Dockerfile, секрети збірки або журнали), щодо якого може бути здійснено шкідливу дію. Суб’єктом є зовнішній зловмисник або розробник, що випадково чи навмисно порушує політику безпеки. Для кожної загрози визначено потенційні наслідки – від вбудовування шкідливого коду у контейнер до компрометації облікових даних чи витоку конфіденційної інформації через логування.

Ймовірність і вплив оцінено якісно на основі частоти виникнення подібних інцидентів у практиці DevSecOps та даних з аналітичних досліджень безпеки контейнерних середовищ. Висока ймовірність та критичний рівень впливу характерні для загроз, пов’язаних із відсутністю контролю запуску збірки та перевірки агентів. Такі загрози здатні призвести до безпосередньої компрометації процесу розгортання. Загрози середньої ймовірності, зокрема витік секретів через Dockerfile або журнали, мають дещо менший вплив, але все одно становлять істотний ризик для цілісності та конфіденційності середовища.

Таким чином, таблиця «Аналіз загроз» відображає системну вразливість Docker builder, пов’язану з відсутністю розмежування дозволених операцій на рівні користувача. Ці вразливості стали основою для подальшого формування вимог до системи ідентифікації та контролю доступу, реалізованих через інтеграцію зовнішнього провайдера ідентичності (IdP). Саме з цих загроз випливають критерії безпеки та інтегрованості, використані у подальшій методу вибору IdP.

Формування вимог до провайдера IDP

У межах методу оцінювання провайдерів ідентичності для інтеграції з Docker builder усі критерії згруповано за п’ятьма ключовими напрямами, кожен із яких має власну вагу у загальному підсумку. Такий підхід забезпечує баланс між безпекою, інтеграційною сумісністю, операційною ефективністю, масштабованістю та економічною доцільністю впровадження. Основна логіка вагового розподілу побудована на специфіці контейнеризованого середовища, де основний акцент робиться на захисті збірки та контролі доступу агентів.

Найважливішою групою є безпека (SEC) з базовою вагою 0,035. Вона охоплює усі аспекти автентифікації, авторизації та моніторингу доступу. Рівень безпеки визначається через п’ять показників, серед яких особливе значення мають посилення OIDC/OAuth2 (через механізми PKCE, nonce, state, token rotation), підтримка багатофакторної автентифікації (TOTP, WebAuthn, FIDO2), гнучкість політик на основі RBAC, ABAC або REBAC, можливість швидкої ревокації токенів і використання короткоживучих облікових даних, а також наявність системи аудиту подій із підтримкою журналів і підписів, придатних для інтеграції із SIEM. Сукупність цих характеристик визначає, наскільки IdP може запобігати несанкціонованим діям у процесі збірки контейнерів.

Друга група критеріїв – інтегрованість (INT) з базовою вагою 0,025 – оцінює здатність провайдера ефективно взаємодіяти з інструментами CI/CD і Kubernetes. Вона відображає, наскільки IdP може бути вбудований у життєвий цикл користувачів і сервісних акаунтів через SCIM або API, чи підтримує він видачу короткоживучих підписаних токенів і сертифікатів для агентів збірки, наскільки коректно реалізовано мапінг claims у політики доступу для Gatekeeper, OPA, ArgoCD або GitLab/Jenkins, а також чи є можливість контролювати операції з реєстром образів (push, pull, delete, tag) через ті самі атрибути доступу. Ця група критеріїв визначає рівень зрілості інтеграції IdP з автоматизованими процесами DevSecOps.

Третя група, керованість і операційні витрати (OPS), має вагу 0,015 і характеризує практичну сторону адміністрування системи. Вона враховує простоту конфігурації, можливість керування через інструменти інфраструктури як коду (Terraform, Helm, Operator), надійність процесів оновлення, застосування патчів, резервного копіювання та відновлення без простоїв, а також доступність якісної документації, підтримки та активного ком’юніті. У

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

DevSecOps-середовищах цей набір показників визначає реальні витрати на експлуатацію і стабільність роботи.

Четверта група – масштабованість і надійність (REL) з вагою 0,015 – оцінює технічну стійкість рішень. У ній враховуються гарантії SLA, можливість геореплікації, підтримка disaster recovery-сценаріїв, рівень продуктивності та затримок при автентифікації під навантаженням, а також наявність перевірених reference-архітектур для Kubernetes і CI. Ці параметри визначають, наскільки обраний провайдер здатен стабільно функціонувати у масштабованих або багатокластерних інфраструктурах.

Остання група – вартість володіння (TCO) із базовою вагою 0,010 – включає економічні аспекти впровадження та утримання системи. Вона охоплює прозорість ліцензування і прогнозованість витрат, початкову вартість інтеграції та міграції з наявних систем, а також сумарні витрати на експлуатацію протягом року. Ці показники дають змогу оцінити не лише безпекову, а й фінансову ефективність рішення.

Кожен підкритерій оцінюється за шкалою від 0 до 5, після чого нормується до діапазону 0–1 поділом на 5. Якщо деякі критерії мають особливо важливе значення для Docker builder, наприклад контроль короткоживучих облікових даних агентів (SEC4, INT2), їхня вага може бути подвоєна відносно інших усередині відповідної групи. Такий підхід дозволяє адаптувати модель під конкретні вимоги безпеки та забезпечити об’єктивність і гнучкість при оцінюванні різних IdP-рішень. Зв’язок між факторами оцінювання IdP і загрозами, наведеними у таблиці, є системним, оскільки кожна група критеріїв безпосередньо відповідає певним типам ризиків, що виникають у процесі збірки Docker-образів. Методологія STORE дає змогу відобразити ці відповідності через призначення вимог безпеки до компонентів системи, і саме через критерії оцінювання можна перевести виявлені загрози у практичні параметри вибору провайдера ідентичності.

Група безпеки (SEC) пов’язана насамперед із загрозами несанкціонованого запуску збірки, використання неперевірених агентів та витоку секретів. Підкритерії SEC1–SEC5 дають змогу оцінити, наскільки IdP здатен мінімізувати ці ризики через контроль автентифікації та авторизації. Зокрема, OIDC/OAuth2 hardening запобігає підробці сесій і викликів збірки, багатофакторна автентифікація унеможливлює використання викрадених облікових даних, гнучкі політики RBAC/ABAC обмежують привілеї користувачів, а підтримка короткоживучих токенів безпосередньо зменшує ризик повторного використання ключів доступу агентами. Наявність аудит-логів і журналів подій забезпечує відстежуваність усіх дій, що важливо при розслідуванні несанкціонованих збірок або витоків конфіденційної інформації.

Група інтегрованості (INT) має прямий зв’язок із загрозами, пов’язаними з CI/CD агентами та процесом збірки. SCIM або API-керування життєвим циклом користувачів і сервісів дозволяє вчасно відкликати доступ неперевірених агентів, що зменшує ризик підміни збірки. Видача короткоживучих підписаних токенів дає змогу кожному агенту працювати лише протягом обмеженого часу, після чого його доступ автоматично припиняється. Claims-to-policy mapping та інтеграція з Gatekeeper або ArgoCD гарантують, що збірка може бути виконана лише в межах дозволених політик, що практично усуває можливість несанкціонованого запуску. Крім того, контроль операцій із реєстром дозволяє запобігти публікації або видаленню образів без належних повноважень.

Фактори керованості (OPS) зменшують ймовірність загроз, що походять від помилок адміністрування або недостатнього оновлення системи. Простота конфігурації та підтримка IaC забезпечують стабільність і зменшують ризик людського чинника, коли адміністратори випадково залишають незахищені порти чи привілеї. Регулярні оновлення та бекапи гарантують усунення вразливостей, що можуть бути використані для компрометації хоста через Dockerfile або журнали.

Група масштабованості й надійності (REL) сприяє зниженню впливу інцидентів. Високі параметри SLA і DR-опції забезпечують швидке відновлення після компрометації, а контроль

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

продуктивності зменшує ризик відмов у доступі, які можуть бути наслідком спроб несанкціонованих викликів або DDoS-навантаження під час збірки.

Вартість володіння (TCO), хоча й не є безпосередньо технічним бар'єром для загроз, впливає на довгострокову життєздатність системи безпеки. Провайдер із передбачуваною структурою витрат і прозорим ліцензуванням дозволяє утримувати стабільний рівень оновлень, моніторингу й аудиту, що знижує ймовірність накопичення технічних боргів та появи критичних вразливостей.

Таким чином, усі групи критеріїв у моделі оцінювання є логічним продовженням аналізу загроз: вони перетворюють потенційні ризики, виявлені у STORE-таблиці, на вимірювані характеристики системи. Завдяки цьому вибір IdP стає не лише технічним або економічним рішенням, а й механізмом управління безпековими ризиками у середовищі Docker builder.

Формування таблиці з результатами оцінювання

Провайдери, що отримали позначку «не підходить», мають критичні прогалини саме для сценарію контрольованої збірки контейнерів. Наприклад, FreeIPA та OpenLDAP добре працюють як каталоги користувачів, але не мають вбудованого OIDC із короткочасними токенами та аудитом дій агентів, що необхідно для захисту від несанкціонованої збірки. Google Identity Platform орієнтована на мобільні й веб-застосунки, але не дає достатньої гнучкості для RBAC/ABAC і прив'язки до CI/CD пайплайнів.

AWS Cognito, Microsoft Entra ID і подібні хмарні рішення можуть бути використані, якщо вся інфраструктура збірки контейнерів уже жорстко прив'язана до відповідної хмари, однак для гібридного середовища з незалежними build-агентами вони не забезпечують потрібної гнучкості.

Таким чином, для середовища з Docker builder та високими вимогами до DevSecOps найбільш придатними залишаються Keycloak, FusionAuth, Ory та комерційні рішення рівня Okta за умови правильної конфігурації й готовності інвестувати в підтримку або ліцензії (табл. 2).

Аналіз результатів

Дані перерахунку змінили лідерів: найвищі підсумки мають Okta 1.87, Azure AD 1.86 та Auth0 1.83. Їхній успіх пояснюється сукупністю п'ятірок у найвагоміших для моделі блоках безпеки й інтеграції. У цих трьох рішень практично без провалів закриті OIDC-hardening, широкий спектр MFA, гнучкі політики доступу, ревокація і короткоживучі креденшали, а також SCIM/API й claims-to-policy. Azure AD додає сильні показники надійності та обслуговування, тож навіть помірніші бали за TCO не впливають на загальний результат, бо ваги витрат у моделі невеликі.

Середня група – Ory 1.70, Keycloak 1.60 і FusionAuth 1.56 – набрала менше через точкові просідання саме там, де зосереджена найбільша вага. В Ory нижчі оцінки за MFA і частиною інтеграцій з життєвим циклом облікових записів, що зменшує суму, попри сильні політики та ревокацію.

Keycloak утрачає бали у зручності адміністрування й оновленнях, а також не завжди має максимальні оцінки за MFA та контроль реєстру; водночас він компенсує це дуже вигідним TCO. FusionAuth виглядає рівно, але без «максимумів» у SEC/INT, через що у зваженій сумі поступається лідерам.

Нижчі підсумки в Google Identity 1.49 і AWS Cognito 1.46 зумовлені тим, що ці сервіси краще відповідають базовим SSO-сценаріям, ніж ролі ядра DevSecOps-контролю збірки. Недобір у гнучкості політик, управлінні короткоживучими агентними токенами та контролі операцій реєстру безпосередньо б'є по найважливіших для моделі критеріях. FreeIPA 1.11 опинився останнім через структурні обмеження: каталожна природа й слабша підтримка сучасних протоколів і інтеграцій не дозволяють набрати бали в SEC та INT, хоч витрати й виглядають привабливо.

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

Таблиця 2

Результати оцінювання провайдерів

Вимога	Критерій	Провайдер								
	Вага (w_i)	Keycloak	Auth0	Okta	FusionAuth	Ory	AWS Cognito	Azure AD	Google Identity	FreeIPA
SEC1 OIDC hardening	0.035	0.14	0.175	0.18	0.14	0.18	0.14	0.175	0.14	0.07
SEC2 MFA спектр	0.035	0.105	0.175	0.18	0.14	0.11	0.14	0.175	0.14	0.07
SEC3 RBAC/ABAC гнучкість	0.035	0.175	0.14	0.18	0.14	0.18	0.105	0.14	0.105	0.14
SEC4 Revocation	0.035	0.14	0.175	0.18	0.14	0.18	0.14	0.175	0.14	0.07
SEC5 Аудит / журналювання	0.035	0.14	0.175	0.18	0.14	0.14	0.105	0.175	0.105	0.105
INT1 SCIM / API lifecycle	0.025	0.1	0.125	0.13	0.075	0.08	0.075	0.125	0.075	0.05
INT2 Короткоживучі токени	0.025	0.1	0.125	0.13	0.1	0.13	0.1	0.125	0.1	0.05
INT3 Mapping claims політики	0.025	0.125	0.1	0.1	0.1	0.13	0.075	0.1	0.075	0.075
INT4 Контроль операцій реєстру	0.025	0.125	0.1	0.13	0.1	0.13	0.075	0.125	0.075	0.1
OPS1 Простота адміністрування	0.015	0.045	0.075	0.08	0.06	0.05	0.06	0.075	0.06	0.045
OPS2 Оновлення / патчі / бекапи	0.015	0.045	0.075	0.08	0.06	0.06	0.075	0.075	0.075	0.045
OPS3 Документація / підтримка	0.015	0.06	0.075	0.08	0.06	0.06	0.045	0.075	0.06	0.03
REL1 SLA / HA / DR	0.015	0.045	0.075	0.08	0.06	0.06	0.075	0.075	0.075	0.045
REL2 Продуктивність / latency	0.015	0.06	0.075	0.08	0.06	0.06	0.075	0.075	0.075	0.045
REL3 Reference-архітектури	0.015	0.075	0.06	0.08	0.06	0.08	0.06	0.075	0.045	0.045
TCO1 Ліцензування / прогноз	0.01	0.05	0.03	0.02	0.04	0.05	0.04	0.03	0.05	0.05
TCO2 Вартість впровадження	0.01	0.03	0.04	0.03	0.04	0.03	0.03	0.03	0.04	0.02
TCO3 Вартість експлуатації	0.01	0.04	0.03	0.02	0.04	0.04	0.04	0.03	0.05	0.05
Score		1.6	1.825	1.87	1.555	1.7	1.455	1.855	1.485	1.105

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317

Висновки

У результаті проведеного дослідження сформовано метод вибору постачальника системи керування ідентичністю (IdP) для інтеграції в середовище Docker builder з метою запобігання виконанню несанкціонованих операцій. На основі аналізу загроз, визначених у межах підходу Security Threat Oriented Requirements Engineering (STORE), було побудовано систему критеріїв і ваг, що відображають вплив кожного аспекту IdP на усунення ризиків, пов'язаних із неконтрольованими діями розробників, агентів збірки та компрометацією секретів. Оцінювання провайдерів за сформованою моделлю показало, що найвищі результати отримали Okta, Azure AD та Auth0, які забезпечують повний набір функцій OIDC-hardening, багатофакторної автентифікації, політик RBAC/ABAC, а також зрілі інтеграційні механізми SCIM та API. Їхні архітектурні особливості дозволяють ефективно впровадити принцип найменших привілеїв і централізований контроль доступу до CI/CD-компонентів. Рішення з відкритим кодом, такі як Keycloak і Ory, демонструють хорошу гнучкість і економічну ефективність, однак потребують додаткової конфігурації для досягнення повної функціональності на рівні безпеки та операційної надійності.

Перелік посилань

1. R. S. Sandhu, E. J. Coyne, F. L. Hal та C. E. Youmank. Role-Based Access Control Models // IEEE Computer, pp. 38-74, 1996.
2. J. Glöckler, J. Sedlmeir, M. Frank та G. Fridgen. A Systematic Review of Identity and Access Management Requirements in Enterprises and Potential Contributions // Business & Information Systems Engineering, pp. 421-440, 2023.
3. V. Radha та D. Sahitha. A Survey on Single Sign-On Techniques // Procedia Technology, pp. 134-139, 2012.
4. A. Zineddine, Y. Belfaik, A. Rehaimi, Y. Sadqi та S. Safi. Single Sign-On Security and Privacy: A Systematic Literature Review // Computers, Materials & Continua, pp. 4019-4054, 2025.
5. R. R, S. Mathi, S. G та S. M. An Empirical Investigation of Docker Sockets for Privilege Escalation and Defensive Strategies // 5th International Conference on Innovative Data Communication Technologies and Application, Coimbatore, 2024.
6. H. M. Kiran та Z. Ali. Requirement Elicitation Techniques for Open-Source Systems: A Review // International Journal of Advanced Computer Science and Applications, т. 9, № 1, pp. 330-334, 2018.
7. J. Elijah, A. Mishra, M. Chukwu Udo, A. Abdulganiyu та A. Musa Aibinu. Survey on Requirement Elicitation Techniques: It's Effect on Software Engineering // International Journal of Innovative Research in Computer and Communication Engineering, pp. 9201-9215, 2017.
8. C. Pacheco, I. García та M. Reyes. Requirements elicitation techniques: asystematic literature review based on thematurity of the techniques // IET Software, pp. 365-378, 2018.
9. A. Van Lamsweerde. Goal-Oriented Requirements Engineering: A Guided Tour // EEE International Requirements Engineering Conference (RE), Toronto, 2001.
10. T. Jamal Ansari, D. Pandey та M. Alenezi. STORE: Security Threat Oriented Requirements Engineering // Journal of King Saud University – Computer and Information Sciences, pp. 1210-1211, 2020.
11. R. M. X. WuI, Y. Wang, N. Shafiabady, H. Zhang, W. Yan, J. Gou, Y. Shi, B. Liu, E. Gide, C. Kang, Z. Zhang, B. Shen, X. Li, J. Fan, X. He, J. Soar, H. Zhao, L. Sun, W. Huo та Y. Wang. Using multi-focus group method as an effective tool for eliciting business system requirements: Verified by a case study // PLOS ONE, т. 3, № 18, pp. 1-16, 2023.

Надійшла до редакції (Received): 16.01.2026
Прийнята до друку (Accepted): 17.03.2026
Опубліковано онлайн (Available online): 30.03.2026

© Дарієнко Д.Г., Когут Н.М. Розробка методу вибору IDP провайдера для інтеграції з Docker. Сучасний захист інформації, 1(65), 21–30.
https://doi.org/10.31673/2409-7292.2026.010317