prompt injection within multi-agent systems. У Lecture notes in computer science (с. 511–520). Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-16092-8_28

3. AlSobeh, A., Gwarzo, Z., & Shatnawi, A. (2025). ShadowPlay: Engineering defenses against role-based prompt injection and dependency hallucination in llm-powered development. У 2025 international conference on cybersecurity and ai-based systems (cyber-ai) (с. 317–325). IEEE. https://doi.org/10.1109/cyber-ai66431.2025.11233258
4. Shi, J., Yuan, Z., Tie, G., Zhou, P., Gong, N., & Sun, L. (2026). Prompt injection attack to tool selection in LLM agents. У Network and distributed system security symposium. Internet Society. https://doi.org/10.14722/ndss.2026.230675

## Метод попарного порівняння AHP для пріоритезації безпекових контролів SSDF у CI/CD

УДК 004.056:004.4

Тарас Лечаченко[1], Дмитро Войтович[2]

*Тернопільський національний технічний університет імені Івана Пулюя,*
*[1]lechachenko.taras@tntu.edu.ua, [2]voitovuch855@gmail.com*

У зв'язку зі зростанням кількості кіберзагроз і складністю CI/CD-інфраструктур особливої актуальності набуває завдання пріоритизації безпекових контролів для DevSecOps та CI/CD-середовищ із подальшим кількісним оцінюванням їхньої ефективності. Існуючі підходи переважно зосереджуються на виявленні загроз або описі окремих механізмів захисту, однак недостатньо уваги приділяється формалізованим методам оцінювання та ранжування безпекових заходів відповідно до рівня критичності загроз.Як зазначають автори роботи [1], за останні п'ять років кількість досліджень у сфері DevSecOps суттєво зросла, проте питання пріоритезації безпекових контролів у контексті конкретних загроз для CI/CD-інфраструктури залишається недостатньо дослідженим.

Одним із перспективних підходів до розв'язання цієї задачі є застосування методу аналізу ієрархій (AHP, Analytic Hierarchy Process) [2], який базується на попарному порівнянні критеріїв та альтернатив. Використання AHP дозволяє формалізувати процес прийняття рішень, визначити вагомість окремих загроз і безпекових контролів, а також забезпечити обґрунтовану пріоритизацію заходів захисту для DevSecOps та CI/CD-середовищ. Для ранжування заходів захисту проти загроз CI/CD за основу взято STRIDE-методологію та визначено, що серед шести категорій STRIDE три в контексті CI/CD є першочерговими: підробка коду та артефактів, підміна сутності та розкриття інформації. Безпекові контролі для ідентифікації засобів захисту конвеєрів CI/CD було взято з NIST Secure Software Development Framework (SSDF) методології [3], оскільки вона забезпечує орієнтовані на життєвий цикл практики, що відповідають DevSecOps. Для забезпечення точності та практичної доцільності пріоритезації з переліку контролів SSDF було обрано підмножину з десяти засобів контролю:

- PO.1 — Визначення вимог безпеки для програмного забезпечення;
- PO.3 — Захист програмного забезпечення від відомих вразливостей;
- PW.1 — Ідентифікація та захист конфіденційних даних;
- PW.2 — Імплементація принципу найменших привілеїв;
- PW.4 — Безпечне зберігання та керування обліковими даними;
- RV.1 — Перевірка архітектури програмного забезпечення з точки зору безпеки;
- RV.2 — Перевірка коду для виявлення вразливостей безпеки;
- RV.3 — Перевірка програмного забезпечення на наявність логування та аудиту;
- RV.4 — Верифікація цілісності програмного забезпечення;
- PO.4 — Безперервний контроль та покращення практик.

Відповідно до представлених альтернатив (контролів) троє експертів із досвідом роботи у кібербезпеці понад 5 років здійснили попарне порівняння пріоритетності застосування контролів відносно трьох критеріїв загроз моделі STRIDE: Tampering, Spoofing, Information disclosure. Локальні пріоритети за критерієм Tampering обчислені за допомогою геометричного середнього згідно з методологією AHP для агрегації думок експертів. Відповідно агреговані значення Tampering представлені у таблиці 1.

Таблиця 1

Локальні пріоритети контролів SSDF за критерієм загрози Tampering

| RV.4 | PW.2 | RV.3 | PW.4 | PO.3 | RV.1 | PW.1 | PO.4 | RV.2 | PO.1 |
|---|---|---|---|---|---|---|---|---|---|
| 0.266 | 0.184 | 0.168 | 0.101 | 0.071 | 0.069 | 0.048 | 0.041 | 0.026 | 0.020 |

Як і індивідуальні оцінки експертів, так і агрегована матриця Tampering є узгодженою CR=0,05 (для 10 альтернатив RI (random index) = 1.49). Результати пріоритезації за критерієм загрози Tampering показали, що найважливішими контролями є RV.4 — «Верифікація цілісності програмного забезпечення» (0.266), PW.2 — «Імплементація принципу найменших привілеїв» (0.184) та RV.3 — «Перевірка програмного забезпечення на наявність логування та аудиту» (0.168), оскільки саме вони безпосередньо спрямовані на запобігання несанкціонованій модифікації коду та артефактів у CI/CD-конвеєрі. Середній рівень пріоритету отримали PW.4, PO.3 та RV.1, що забезпечують додатковий захист через керування обліковими даними, виявлення вразливостей і аналіз архітектури безпеки. Найменші значення отримали PW.1, PO.4, RV.2 та PO.1, оскільки їх вплив на протидію Tampering є більш опосередкованим.

Наведений приклад пріоритизації є лише фрагментом більш комплексного AHP-розрахунку, виконаного в межах дослідження. Повний аналіз включає розрахунок локальних і глобального пріоритетів засобів контролю SSDF не лише для загрози Tampering, але й для категорій Spoofing та Information Disclosure. Отримані результати підтверджують дієвість запропонованого

підходу в умовах обмежених ресурсів та необхідності визначення найбільш критичних заходів захисту для CI/CD-середовищ.

1. Prates, L., & Pereira, R. (2025). DevSecOps practices and tools. International Journal of Information Security, 24 (1), 11. https://doi.org/10.1007/s10207-024-00914-z
2. Saaty, T. L., & Vargas, L. G. (2012). Models, Methods, Concepts & Applications of the Analytic Hierarchy Process (2nd ed.). Springer. DOI: doi.org/10.1007/978-1-4614-3597-6
3. National Institute of Standards and Technology. (2022). Secure Software Development Framework (SSDF) version 1.1: Recommendations for mitigating the risk of software vulnerabilities (NIST Special Publication 800-218). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-218

## Кібербезпека систем екологічного моніторингу як елемент критичної міської інфраструктури

УДК 004.056:502.3:004.738.5

Андрій Станько[1], Ірина Дідич[2], Артем Гончаренко[3]

*Тернопільський національний технічний університет імені Івана Пулюя,*
*[1]andrjj.stanko@gmail.com, [2]iryna.didych1101@gmail.com*
*Київський національний університет будівництва і архітектури,*
*[3]hosting.pat@gmail.com*

Системи екологічного моніторингу є важливим елементом цифрової інфраструктури міста, адже забезпечують збирання, передавання, оброблення та візуалізацію даних про якість повітря, стан води, шумове навантаження, радіаційний фон, мікроклімат і екологічні ризики. У smart city вони поєднуються з IoT-пристроями, геоінформаційними платформами, диспетчерськими службами та публічними панелями, тому достовірність і доступність даних впливають на управлінські рішення та реагування служб.

Актуальність проблеми зумовлена тим, що кібератаки на такі системи можуть спричинити втрату даних або формування хибної картини екологічної ситуації. Підміна телеметрії, блокування панелей, компрометація хмарної платформи чи втручання в алгоритми оброблення підвищують ризик запізнілого реагування на аварійні викиди, пожежі та техногенні інциденти. Метою роботи є обґрунтування підходу до кіберзахисту систем екологічного моніторингу як елемента критичної міської інфраструктури.

Відповідно до NIST Cybersecurity Framework 2.0, управління кібербезпекою охоплює ідентифікацію активів і ризиків, захист, виявлення подій, реагування, відновлення та організаційне управління [1]. Для екологічного моніторингу об’єктом захисту є повний ланцюг формування інформації: сенсор, контролер, канал зв’язку, шлюз, edge-вузол, API, хмарна платформа, аналітичний модуль, інтерфейс оператора і публічний вебсервіс.