Аналіз проблем кібербезпеки при використанні програмного забезпечення з відкритим кодом

УДК 004.052.2

Андрій Тарасенко[1], Мар'ян Кирик[2]

Національний університет "Львівська політехніка",
[1]andrii.d.tarasenko@lpnu.ua, [2]marian.i.kyryk@lpnu.ua

Програмне забезпечення (ПЗ) з відкритим кодом набуло широкого використання за рахунок надійності, гнучкості і невеликих фінансових вкладень. Операційні системи (ОС) на основі відкритого ядра Linux займають вагому частку серверної інфраструктури та досить часто є основним вибором для розгортання веб-серверів, баз даних, хмарних платформ та інших важливих систем. Однак ці ОС несуть певні архітектурні та безпекові ризики, оскільки супроводом розробки займається не конкретна компанія, а публічна спільнота розробників.

Метою даної роботи є аналіз проблем та методів покращення захисту критичної IT інфраструктури при використанні програм з відкритим кодом.

Тільки за перші три місяці 2024 було виявлено декілька критичних вразливостей:

- CVE-2024-23897 Вразливість дозволяє зловмиснику, який не пройшов перевірку автентичності, виконувати віддалене виконання коду в системі Jenkins. [1]

- CVE-2024-0402 Вразливість дозволяє віддаленому зловмиснику, який пройшов перевірку автентичності, виконувати запис довільних файлів на сервері GitLab при створенні робочого простору. [2]

- CVE-2024-3094 у пакеті XZ, виявлено бекдор який може бути використаний для віддаленого доступу до системи де встановлений SSH сервіс без авторизації. [3]

Ці та інші випадки які час від часу виявляються в програмах з відкритою кодовою базою стають серйозним прецедентом для перегляду забезпечення кібербезпеки при використанні таких програм у критичній інфраструктурі. Проблематика навколо цих вразливостей піднімає більш глибокі питання щодо безпеки у проектах відкритого програмного забезпечення.

Аналізуючи проблему з нещодавно виявленою вразливістю у пакеті XZ можна відмітити що зловмисникам необов'язково використовувати складні методи компрометації. В даному випадку було достатньо отримати статус довіреної особи проекту шляхом внесення незначних правок в кодову базу. Після чого зловмисники замаскували бекдор під типову зміну в коді і розробник, що займається супроводом проекту не запідозривши нічого додав його до основної гілки програми. Разом з тим було сформовано пакети інсталяції програм для більшість дистрибутивів на основі ядра Linux, які в подальшому були встановлення вже в операційних системах по всьому світу.

Універсальний метод для захисти ПЗ від такого виду атак, нажаль, відсутній. Проте, пропонується сформувати основні підходи для мінімізація ризиків та стратегій протидії у випадку їх виявлення:

1) Ретельний вибір проектів з відкритим кодом. Рекомендується використовувати операційні системи з відкритим кодом, підтримкою яких займаються великі компанії такі як: RedHat, SUSE, Ubuntu Enterprice. Вони

проводять аудит кодової бази та несуть відповідальність та випускають оновлення безпеки за їх необхідності.

2) Підтримка балансу між функціональністю та надійністю. Більшість програм, як правило, мають декілька редакцій: одну з довготривалою підтримкою (Long Time Support – LTS) і одну з останнім оновленням (Latest). В критичній інфраструктурі рекомендується використовувати саме програми з LTS, оскільки вони призначенні для гарантованої роботи та містять перевірені та протестовані оновлення [4].

3) Оскільки ніхто не застрахований від помилок, потрібно постійно відслідковувати виявлення критичних вразливостей та вчасно проводити оновлення ПЗ.

4) Рекомендується залучення штучного інтелекту для перевірки кодової бази проектів[5].

5) Використовувати заходи по обмеженню прав доступу не лише користувачів, а також сервісів та ПЗ.

6) Використовувати інструменти моніторингу трафіку та завантаженості системи; централізованого збору і аналізу журналів подій для відслідковування небажаної активності та вчасної реакції на нетипові інциденти.

Таким чином, провівши аналіз проблем та методів покращення захисту критичної IT інфраструктури при використанні програм з відкритим кодом, рекомендується застосовувати комплексний підхід для покращення кібербезпеки, надавати перевагу тільки перевіреному ПЗ, безперервно відслідковувати виявлені критичні вразливості та обмежувати доступ не тільки на рівні користувача, але й на рівні програм та сервісів, що в свою чергу дозволить мінімізувати внутрішні та зовнішні кіберзагрози в інфраструктурі.

1. “Jenkins Security Advisory 2024-01-24" URL: https://www.jenkins.io/security/advisory/2024-01-24/ (дата звернення 15.03.2024).
2. “GitLab Critical Security Release: 16.8.1, 16.7.4, 16.6.6, 16.5.8” URL: https://about.gitlab.com/releases/2024/01/25/critical-security-release-gitlab-16-8-1-released/ (дата звернення 15.03.2024).
3. “CVE-2024-3094" URL: https://ubuntu.com/security/CVE-2024-3094 (дата звереннння 02.04.2024).
4. Calles M., Serverless Security: Understand, Assess, and Implement Secure and Reliable Applications in AWS, Microsoft Azure, and Google Cloud, 1st ed., NY, USA, Apress, 2020 pp 39-64.
5. A. Rodrigo, B. Paulo, “Privacy and security constraints for code contributions”, Software - Practice and Experience, vol. 50, issue 10, pp 1905 – 1929, October 2020.