

УДК 004.056.5

Павленко К.Є.¹, Зайко Т.А.²

¹ студ. гр. КНТ-124м НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ПРИНЦИПИ «БЕЗПЕЧНОГО» КОДУВАННЯ

Сучасний світ нерозривно пов'язаний із використанням програмного забезпечення, яке забезпечує роботу різних систем і послуг. Однак цей зростаючий попит на програми також призводить до збільшення кількості потенційних загроз для інформаційної безпеки та конфіденційності даних користувачів. У цьому контексті принципи «безпечного» кодування набувають надзвичайної важливості.

Безпечне програмування — одна з форм безпечного проектування програм, мета якої забезпечити тривале функціонування певної частини коду програми під впливом непередбачуваних обставин [1].

Вимоги до розробки програмного забезпечення, як правило, орієнтовані на функціонал і наголошують на конструктивних аспектах, так що деструктивний погляд на потенційні вразливості безпеки здебільшого діаметрально протилежний реальним цілям розробки: багатьом програмістам просто бракує структурованого і загостреного погляду на кібербезпеку [2].

Основні принципи безпечного кодування в програмуванні спрямовані на мінімізацію ризику вразливостей і атак на програмне забезпечення. Ось декілька основних принципів безпечного кодування:

- мінімізація атакуючої поверхні. Зменшення можливостей для атак шляхом обмеження функцій та можливостей, які доступні користувачам із зовнішнього середовища. Провести аналіз і визначити, які частини програми можуть бути доступні для потенційних атак і обмежити їх;

- валідація введення. Перевірка і фільтрація всіх введених даних, щоб уникнути атак на введення, такі як SQL-ін'єкції, XSS-атаки і інші. Ніколи не довіряйте даним, які надходять від користувачів або з зовнішніх джерел;

- безпечність аутентифікації та авторизації. Забезпечення надійних методів аутентифікації користувачів і контролю над їх доступом до ресурсів. Важливо визначити, хто має доступ до яких функцій і даних із суворим дотриманням прав доступу;

- захист даних. Використання механізмів шифрування та інших методів захисту для збереження конфіденційності та цілісності даних. Ніколи не зберігайте конфіденційну інформацію в явному вигляді;

- управління помилками та винятками. Обробка помилок та винятків повинна бути належним чином налаштованою, щоб уникнути розкриття чутливої інформації та запобігти втраті даних або некоректній роботі програми;

- оновлення та патчі. Забезпечення своєчасного виправлення виявлених уразливостей та надання оновлень для програмного забезпечення з метою підтримки його безпеки в актуальному стані;

- перевірка вразливостей. Регулярне тестування і аудит безпеки програмного забезпечення для виявлення можливих уразливостей і вирішення їх;

- безпека сторонніх компонентів. Перевірка безпеки сторонніх бібліотек і компонентів, які використовуються в програмі, і вчасне оновлення їх для виправлення вразливостей;

- документація. Збереження докладної документації щодо безпечних практик, архітектури програми та інших аспектів безпеки для спрощення спільної роботи та аудиту.

Важливість та необхідність безпечного кодування не можна недооцінювати в сучасному світі. Забезпечення безпеки програмного забезпечення має вирішальне значення для захисту користувачів, конфіденційності даних і запобігання фінансовим втратам та репутаційним ризикам. Принципи безпечного кодування є необхідними для запобігання вразливостям і атакам, що можуть нанести серйозну шкоду як користувачам, так і організаціям. Дотримання цих принципів створює надійне та стійке програмне забезпечення, яке може витримати випробування часом і забезпечити безпеку від сучасних кіберзагроз.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Pevnev V. Analysis and research of well-known orchestration systems for the construction of microservice infrastructure / V. Pevnev, Yu. Trehub. – Advanced Information Systems, 2020. – Vol. 4. – No. 2. – 142 – 147 p.

2. Безпечне кодування - виклики інформаційної безпеки [Електрон. ресурс]. – Режим доступу : <https://www.dqsglobal.com/uk-ua/navchajtesya/blog/bezpechne-koduvannya-vikliki-informacijnoyi-bezpeki#grundsatzeder-sicheren-Codierung-chapter04>.