ОГЛЯД

Зі збільшенням складності та обсягу сучасних програмних систем зростають і виклики у пошуку та виправленні дефектів програмного забезпечення, зокрема вразливостей. Традиційні методи, такі як інструменти статичного аналізу коду та ручні перегляди коду, зазвичай є неефективними для великих і складних кодових баз. Ці підходи зазвичай дають високий рівень хибнопозитивних результатів і не можуть виявляти вразливості під час виконання, що означає, що критичні дефекти залишаються невиявленими, а безпека системи порушується.

Це дослідження пропонує рішення проблеми за допомогою представлення моделі передбачення дефектів VulGraphNet, яка використовує комбінований підхід статичного та динамічного аналізу з використанням графових нейронних мереж. Ця модель ґрунтується на багаторівневому графічному представленні: графах потоку керування, графах потоку даних і абстрактних синтаксичних деревах, щоб у складний спосіб відображати взаємозалежні зв’язки в коді. VulGraphNet надає більш комплексне рішення для виявлення дефектів, що виникають через складні взаємодії між різними компонентами програмного забезпечення, інтегруючи статичний аналіз, який виявляє структурні вразливості в коді, і динамічний аналіз, який моніторить поведінку під час виконання.

Серед основних досягнень цього дослідження - розвиток методик машинного навчання для інтеграції графових нейронних мереж з метою покращення вилучення ознак. GNN природно підходять для аналізу даних, структурованих у вигляді графів, що дозволяє моделі навчатися складним шаблонам і асоціаціям між різними частинами програми. Це глибоке навчання ознак дозволяє виявляти потенційні уразливості безпеки, які можуть залишатися непоміченими за допомогою традиційних підходів, особливо ті, що виникають через складні залежності та поведінки, чутливі до контексту. Крім того, VulGraphNet використовує техніки багатовимірного злиття ознак, що підвищує здатність моделі до узагальнення і робить її застосовною до більшої кількості програмних проектів - від маломасштабних систем до великих розподілених додатків.

У цьому дослідженні використовуються кілька публічно доступних наборів даних вразливостей для проведення експериментів на запропонованій моделі VulGraphNet. Результати показали відмінні результати як за точністю виявлення, так і за охопленням. Це стало можливим завдяки комбінованому використанню статичного та динамічного аналізу, що зменшує кількість хибнопозитивних і хибнонегативних результатів, що в свою чергу збільшує точність і відновлення при виявленні дефектів. Це призводить до найбільш

ефективного та надійного методу передбачення вразливостей, що є дуже важливим процесом у розробці програмного забезпечення, що стосується порушень безпеки та зменшення витрат на обслуговування.

Результати цього дослідження підкреслюють важливість інтеграції різних методів аналізу, що є необхідним для подолання обмежень, виявлених у сучасних методологіях виявлення вразливостей. Інтеграція динамічного аналізу з статичним аналізом, заснованим на машинному навчанні, дозволяє створити більш надійну систему для виявлення складних вразливостей, що залежать як від структури, так і від поведінки коду під час виконання. В кінцевому рахунку, це дослідження робить внесок у розвиток безпеки програмного забезпечення, надаючи точний, ефективний і масштабований інструмент для виявлення дефектів на ранніх етапах.

На завершення, VulGraphNet представляє значне вдосконалення у виявленні вразливостей програмного забезпечення, використовуючи переваги статичного та динамічного аналізу, графових нейронних мереж і багатовимірного злиття ознак для точніших і масштабованих систем виявлення вразливостей. Це дослідження сприятиме не лише підвищенню ефективності виявлення дефектів програмного забезпечення, але й прокладе шлях для подальших досліджень у галузі автоматизованих інструментів для забезпечення безпеки програмного забезпечення.

Ключові слова: виявлення вразливостей програмного забезпечення, графові нейронні мережі (GNN), статичний аналіз, графові мережі з увагою (GAT).