УДК 004.6 : 004.62

Магістрант Шкільнюк В. О., доктор філософії Молчанов О. А.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

КОМП'ЮТЕРНА СИСТЕМА АСИНХРОННОГО ФІЛЬТРУВАННЯ МЕРЕЖЕВИХ ПАКЕТІВ

Abstract

Shkilniuk Vladyslav, master student; Oleksii Molchanov, PhD

Computer system for asynchronous network packet filtering

This paper explores the relevance of packet filtering nowadays and its specifics. A comprehensive review and analysis of existing solutions, including static packet filters are conducted. The study proposes an asynchronous system based on reactor pattern that allows executing multiple tasks in a single threaded event loop to maximize performance. Potential ways for future enhancements are discussed.

Вступ

Комп'ютерні мережі дають змогу різним пристроям обмінюватись інформацією та ресурсами. Для забезпечення захисту та цілісності даних в мережах використовуються різні засоби фільтрування та контролю трафіку.

Пакетні фільтри — одні з найстаріших, проте не менш важливі елементи захисту мереж. Аналізуючи головні складові повідомлень — такі як IP адреса, порт, протокол, вони приймають рішення чи варто пропускати мережевий пакет для подальшого опрацювання чи слід відкинути його на ранньому етапі і не витрачати системні ресурси.

Навіть зараз пакетні фільтри залишаються актуальними через свою простоту та швидкість. Більш складні системи безпеки такі як брандмауери базуються на пакетних фільтрах.

Постановка задачі

Метою даного дослідження є розробка асинхронної системи фільтрування мережевих пакетів з можливістю звітування про поточну конфігурацію та кількість відфільтрованих пакетів за рахунок використання циклу подій і механізму асинхронного читання з сокета. Відповідно до вказаної мети необхідно розв'язати такі задачі: оглянути наявні мережеві фільтри, проаналізувати основу їх роботи, розробити асинхронну та адаптивну систему фільтрування мережевих пакетів,

перевірити її ефективність та проаналізувати можливості подальшого вдосконалення.

Термінологія

User space (користувацький простір) — контекст виконання користувацьких програм, що знаходиться над рівнем ядра.

Сокет — інтерфейс для забезпечення обміну даними між процесами, системами, програмами.

Kernel space — контекст виконання ядра операційної системи.

Poll/epoll — механізми операційної системи, призначенні для вводу/виводу. Вони дають змогу програмам очікувати події на кількох файлових дескрипторах одночасно.

Аналіз наявних підходів до фільтрування мережевих пакетів

Наявні системи фільтрування пакетів — такі, як традиційні брандмауер, user space фільтри на базі традиційних пакетних фільтрів iptables, nftables [1] — зазвичай використовують статичні набори правил. Ці правила дуже часто фіксовані, виконуються послідовно і рідко адаптуються до мережевих змін.

Утиліта iptables працює як user space конфігуратор правил для kernel space фреймворку для фільтрування пакетів. Вона реалізована як машина станів, яка перевіряє кожен пакет згідно наявних правил, для прийняття рішень про те, чи треба пропускати пакет далі. В момент очікування iptables не робить ніякої корисної роботи. Так як iptables перевіряє пакет послідовно відносно кожного правила, то з’являється залежність від кількості правил, що впливає на кількість часу на опрацювання кожного пакета.

Утиліта nftables — більш новий пакетний фільтр, який прийшов на заміну iptables. До ключових відмінностей від iptables слід віднести уніфікований інтерфейс для роботи з різними протоколами, використання таких структур даних, як множини та словники для підвищення продуктивності. Але nftables реалізовує ту ж саму машину станів і використовує статичні правила.

Наявні фільтри можна вважати асинхронними у сенсі того, як вони очікують на вхідні дані, проте час очікування ніяк не використовують.

Опис системи

Для створення асинхронної системи фільтрування мережевих пакетів пропонується використання циклу подій бібліотеки Boost asio [3] замість машини станів, що дає змогу одночасно використовувати багато асинхронних операцій вводу-виводу без блокування основних функцій.

Основою архітектури є паттерн реактор [2] (рис. 1). Для виключення блокувань при читанні з сокета використовуються такі механізми операційної системи Linux, як poll та epoll.

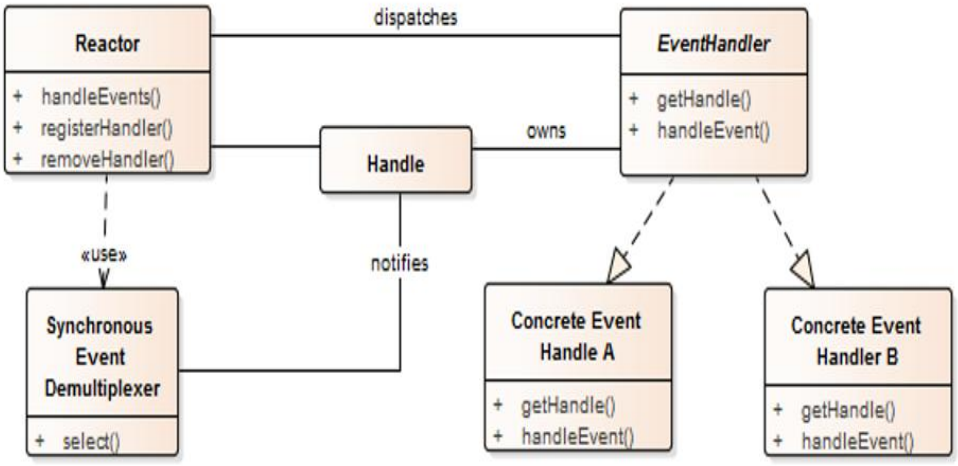

Рис. 1. Паттерн реактор

В результаті це дасть змогу використати час очікування мережевих пакетів для виконання корисних операцій, наприклад опрацювання наявних пакетів. Для класифікації мережевих пакетів пропонується модель на базі ймовірнісного алгоритму Баєса [5], що описується формулою:

$$P(C \vee x) = \frac{P(x \vee C) * P(C)}{P(x)} \quad (1)$$

Де x — ознака пакета така, як порт, розмір пакета, протокол, наявність прапорців, наприклад SYN TCP протоколу. Це дасть змогу виявляти різку зміну мережевого трафіку. Актуальність попередніх даних забезпечується механізмом ковзного вікна, що обмежує кількість останніх спостережень. В якості основних характеристик пакета можна обрати діапазон портів, протокол, у випадку TCP можна виділити окрему ознаку — присутність лише SYN біта.

Так як асинхронні системи легко розширюються з додаванням інших операцій, то основні функції фільтрування пропонується розширити механізмом звітування про поточний стан системи у вигляді асинхронного сервера, що буде працювати в межах системи, і можливість змінювати конфігурацію системи в реальному часі через окремий файл і асинхронне очікування на зміни у ньому, використовуючи inotify [4].

Результати експериментальних досліджень

Реалізацію системи виконано мовою C++ з використанням бібліотеки Boost asio. Для перевірки роботи системи проведено тест, в якому якийсь

час отримували трафік одного типу, наприклад ICMP, а потім TCP. Таким чином, перший час система бачить лише пакети типу ICMP, що формують апріорну ймовірність поточного трафіку, в якому будь-який наступний пакет з протоколом ICMP буде вважатись «нормальним». Коли система зустрічає перший TCP пакет, формуються нові ознаки і апостеріорна ймовірність таких пакетів буде низькою, що дає можливість детектувати різку зміну трафіку. Так як класифікатор Байєса оперує, виходячи з того факту, що ознаки незалежні, то система буде помічати різку зміну пакетів по окремим ознакам. В момент роботи системи, було декілька разів отримано звіт та змінено конфігурацію. Всі події були виконані асинхронно.

Висновки

Дана стаття присвячена розробці комп'ютерної системи асинхронного фільтрування мережевих пакетів. Особливістю якої є використання асинхронної моделі фільтрування пакетів на базі циклу подій, що дає можливість використати час очікування для оновлення моделей, конфігурації системи, звітування про результати роботи. Запропонована система дає змогу визначати різку зміну мережевого потоку за короткий проміжок часу за рахунок використання ймовірнісного класифікатору Баєса, який дає змогу визначити, наскільки поточний мережевий пакет відповідає класу «нормальний».

Подальшого дослідження потребують такі питання: можливість реалізації zero-copy для роботи з мережевими пакетами в користувацькому просторі для зменшення накладних витрат при аналізі.

Література

1. Iptables vs nftables in Linux: What is the difference? URL: https://tuxcare.com/blog/iptables-vs-nftables/
2. Паттерн реактор. URL: https://www.modernescpp.com/index.php/reactor/
3. Документація на Boost asio URL: https://www.boost.org/doc/libs/latest/doc/html/boost_asio.html
4. Kerrisk M. The Linux Programming Interface: A Linux and UNIX System Programming Handbook. San Francisco: No Starch Press, 2010. 1552 p.
5. Bayes' Theorem in Machine Learning: A Literature Review URL: https://direct.ewa.pub/proceedings/tns/article/view/20329