УДК 004.414
DOI: 10.31673/2409-7292.2025.015886

Кулик Ю.А., Лах Ю.В.

АНАЛІЗ БЕЗПЕКИ МЕРЕЖЕВИХ ПЛАГІНІВ KUBERNETES

Робота присвячена розгляду Kubernetes, що застосовує стандарт Container Network Interface (CNI) для забезпечення зв'язку між контейнерами, розгорнутими на різних нодах, надаючи гнучкий та масштабований підхід до конфігурації мережі. У статті розглядається, як така гнучкість, будучи корисною для ефективного використання ресурсів та спрощення керування, водночас створює також і певні ризики з точки зору безпеки. Предметом аналізу стають чотири популярні мережеві плагіни — Flannel, Calico, Weave Net та Cilium, розглядається їх архітектура, способи забезпечення зв'язку між подами та підкреслюються різноманітні ризики під час їх використання, як от переміщення зловмисника всередині мережі у разі компроментації поди. Основний акцент зроблено на тому, як різні плагіни реалізують засоби безпеки. Деякі з них надають можливість конфігурування розширених мережевих політик та шифрування трафіку, в той час як інші покладаються на мінімалізм та простоту налаштування. У статті наголошується, що ефективна ізоляція трафіку не може ґрунтуватися лише на налаштуваннях за замовчуванням, особливо з огляду на «плоску» мережеву модель Kubernetes. Саме тому адміністраторам платформи рекомендується поєднувати стратегії сегментації мережі, суворого дотримання принципу найменших привілеїв, регулярного оновлення плагінів та моніторингу середовища. У статті представлено низку рекомендацій що охоплюють технічні заходи, настанови з конфігурації та організаційні процеси. До них належать застосування мережевих політик, увімкнення шифрування, обмеження привілеїв компонентів CNI та їх своєчасне оновлення. У міру розростання платформи Kubernetes, зростає і потреба у ретельному керуванні мережевими плагінами, щоб їх гнучкість та масштабованість не відбувалися за рахунок безпеки

Ключові слова: Kubernetes, CNI, BGP, eBPF, поди, мережеві політики, шифрування трафіку

Вступ та формулювання проблеми

З плином часу Kubernetes, як платформа для оркестрації контейнерів, стала галузевим стандартом, забезпечуючи розгортання програм як у невеликих стартапах, так і на великих підприємствах, не виключаючи державні структури, такі як Міністерство оборони США [1]. Враховуючи таке широке впровадження, безпека даної платформи стала критичною проблемою, яка потребує значної уваги. Це підтверджується результатами опитування від CNCF, де 40% респондентів повідомили, що вони глибоко стурбовані безпекою Kubernetes [2].

Безпека мережевих плагінів (Container Network Interfaces, CNI) Kubernetes, як одного із важливих компонентів даної платформи, має надзвичайний вплив на безпеку всього середовища. Вразливість у плагіні чи впровадження неправильних налаштувань можуть призвести до спуфінгу трафіку, обходу мережевих політик або навіть компроментації нод. Для прикладу, зловмисник, який проникає з мережевого простору імен под в простір імен нод, може спостерігати та маніпулювати мережевими інтерфейсами ноди та її таблицями маршрутизації [3]. Останніми роками виникало багато проблем із безпекою мережі даної платформи, починаючи від звичайних атак 2 рівня моделі OSI (ARP-спуфінг, MITM) до вразливостей ядра (eBPF), що в свою чергу підкреслює необхідність ретельного аналізу безпеки мережі у Kubernetes.

У цій статті розглядається безпека мережевих плагінів Kubernetes (таких як Calico, Cilium, Flannel, та Weave Net) через призму сучасних досліджень. Проводиться аналіз відомих вразливостей, сценаріїв атак та механізмів захисту від них. Також обговорюється як дизайн кожного плагіну впливає на сегментацію мережі та які це несе наслідки для безпеки.

Аналіз останніх досліджень і публікацій

Останніми роками академічні дослідження інтенсивно вивчали безпеку Kubernetes і було опубліковано низку цікавих досліджень:

- *Minna et al. (2021) – Understanding the Security Implications of Kubernetes Networking*: Стаття IEEE про безпеку та конфіденційність Minna et al. підкреслює як конфігурація мережі Kubernetes може спровокувати «несподівані атаки» у разі нерозуміння концепцій її роботи [3]. Для прикладу, вони підкреслили, що CNI плагіни, які використовують мережевий міст для комунікації між подами всередині ноди, є вразливі до класичних атак 2 рівня, таких як ARP-

спуфінг або DNS-спуфінг, якщо поду було взламано. Однак, плагіни які використовують маршрутизацію 3 рівня або тунелювання “IP-in-IP” уникають розгортання под в одній підмережі, що нівелює такий ризик. Автори також наголосили, що успішна компроментація демона CNI плагіна надає зловмиснику привілейований доступ до мережевого стеку ноди, фактично надаючи повний доступ до мережі кластера Kubernetes. Вони також звернули увагу на те, що не варто покладатися на конфігурацію мережі за замовчуванням (“плоска” мережа з можливістю комунікації між подами “всі до всіх”) та користуватися мережевими політиками для обмеження цього доступу, хоча варто враховувати, що мережеві плагіни є відповідальними за роботу даних політик, тому вони можуть стати ціллю атаки для обходу мережевих політик. Стаття також згадує про реальні CVE: уразливість BGP (CVE-2021-26928) у компоненті Calico та помилку верифікатора eBPF (CVE-2021-31440), які дозволяли отримати неавторизований доступ до мережі, що є доказом наявності безпекових проблем як у традиційних рішеннях для маршрутизації, так і у новіших способах вирішення цієї задачі – eBPF.

• *Budigiri et al. (2021) – Network Policies in Kubernetes: Performance Evaluation and Security Analysis*: Автори вивчали мережеві політики в Kubernetes, оцінюючи їх вплив на продуктивність системи та її безпеку [4]. Вони підтвердили, що мережеві політики грають надзвичайно важливу роль у забезпеченні ізоляції трафіку, але їхнє впровадження (для прикладу, через правила iptables чи eBPF програми) потребує певних ресурсів системи та часу на обробку трафіку. Цікаво, що плагіни, які використовують правила iptables (Calico в одному з режимів роботи), спричиняють вищу затримку у мережевому трафіку зі збільшенням кількості правил. З точки зору безпеки, дослідження підтвердило, що без мережевих політик скомпроментована пода може вільно сканувати або атакувати інші поди в мережі. Також було відзначено, що у звичайних мережевих політик відсутній певний функціонал як логування так і правила на основі DNS імен. Однак, провідні плагіни, такі як Cilium та Calico, пропонують розширення, які вирішують дану проблему та дозволяють писати більш функціональні політики.

• *Технічні документи та блоги*: Постачальники хмарних технологій і компанії з кібербезпеки опублікували численні технічні документи та дописи в блогах, присвячені безпеці плагінів CNI. Для прикладу, у дописі “Exploring eBPF Integration with Kubernetes” (Yevle, 2024) обговорюють як CNI на основі eBPF (наприклад, Cilium) підвищують безпеку системи шляхом застосування політик всередині ядра та забезпечення глибокої видимості [5]. Вони зазначають, що eBPF дозволяє динамічну мікросегментацію, що базується на ідентифікації (за допомогою міток подів), і може блокувати несанкціоновані підключення з меншою затримкою, ніж традиційні проксі

Узагальнюючи, нещодавні дослідження фокусуються на таких ключових моментах:

• Сегментація мережі є надзвичайно важливою та нетривіальною

• Провідні плагіни CNI мали вразливості, виявлені останніми роками, підкреслюючи, що помилки розробки трапляються (навіть у добре налагоджених проектах)

• Активно розробляються різноманітні технології, методи та засоби захисту, які покращують та вдосконалюють мережеві плагіни у Kubernetes (наприклад, eBPF, шифрування, посилена автентифікація для протоколів керування)

Мета та завдання дослідження

Враховуючи наведену вище інформацію. виділяються наступні цілі дослідження:

• Оцінити стан безпеки провідних плагінів CNI у Kubernetes

• Порівняти архітектури мережевих плагінів з точки зору безпеки

• Визначити найкращі практики щодо забезпечення їх безпеки

Для досягнення цих цілей завдання дослідження включають огляд найновішої літератури у даному напрямку (статті, матеріали конференцій, офіційна документація), аналіз бюлетенів

© Кулик Ю.А., Лах Ю.В. (2025). Аналіз безпеки мережевих плагінів Kubernetes. Сучасний захист інформації, 1(61), 311–317. https://doi.org/10.31673/2409-7292.2025.015886.

безпеки та CVE, та, врешті решт, аналіз документації провідних мережевих плагінів щодо впроваджених функцій безпеки. Результати виконання цих завдань надано у наступних секціях.

Провідні CNI плагіни та їх безпека

У цій секції проводиться аналіз чотирьох широко поширених плагінів CNI – Calico, Cilium, Flannel та Weave Net, зосереджуючись на їх архітектурі, мережевій моделі та відомих способах забезпечення безпеки.

Calico

Це популярний CNI, який реалізує мережу 3 рівня [6]. Кожній поді призначається IP-адреса, яка маршрутизується в межах кластера. Він зазвичай використовує BGP для розподілу маршрутів між нодами (кожна нода оголошує діапазони IP-адрес своїх под). Такий дизайн означає, що поди на різних нодах можуть комунікувати через маршрутизовані IP-пакети без використання NAT [7]. Calico має передову підтримку мережевих політик Kubernetes, використовуючи iptables або eBPF для застосування правил, які дозволяють або забороняють трафік між подами. Також є можливим налаштування шифрування (IPsec або WireGuard) для трафіку між подами, та інтеграція із провайдерами ідентифікації для покращеного контролю доступів.

Щодо безпеки даного рішення, так як Calico не покладається на мережеві мости для комунікації між подами, скомпроментована пода не може здійснювати ARP спуфінг, щоб перехопити трафік призначений для інших под на тій самій ноді, так як немає комунікації на 2 рівні [3]. А орієнтація плагіну на маршрутизацію IP означає уникнення певних складнощів властивих оверлейним мережам, що у свою чергу зменшує «поверхню атаки» і, відповідно, ризики безпеки. В додаток, Calico має дуже розвинений механізм мережевих політик, що дозволяє обмежувати трафік у просторах імен, діапазонах IP, а також на 7 рівні (з допомогою технології «Service Mesh» або розширених політик безпеки). Використання даного функціоналу дозволяє реалізувати мережу з нульовою довірою (zero-trust) всередині кластера.

Cilium

Цей новіший мережевий плагін розроблений компанією Isovalent та використовує технологію eBPF для реалізації мережі [8]. Замість конфігурування iptables або мережевих мостів, Cilium запускає eBPF програми у різних точках підключення (для прикладу, вхід/вихід TC інтерфейсів або XDP для ранньої обробки пакетів), щоб реалізувати маршрутизацію пакетів, балансування навантаження та застосування мережевих політик на рівні ядра операційної системи. Він також підтримує мережеві політики Kubernetes, та надає користувачам більш розширений варіант цих політик (CiliumNetworkPolicy CRDs). Щодо фільтрації трафіку, вона може відбуватися як на 3 та 4 рівнях, так і на 7 рівні завдяки технології eBPF [7]. А модель безпеки Cilium, що базується на ідентифікації, призначає кожному поду ідентифікатор (на основі міток) і може застосовувати мережеві політики базуючись на ньому. Також є можливість енкапсуляції трафіку між нодами, використовуючи оверлей Geneve.

З точки зору безпеки, застосування eBPF дозволяє уникнути використання привілейованого проксі в просторі користувача, а також це зменшує ризики переповнення буфера, так як програми eBPF перевіряються ядром на проблеми з використанням пам’яті перед їх запуском. В додаток, так як eBPF запускається у межах контексту ядра, фільтрація пакетів та рішення щодо їх маршрутизації відбувається швидко та менш помітно для скомпроментованих под (як тільки програму завантажено в ядро, на неї неможливо вплинути з простору користувачів). Щодо фільтрації трафіку, то навіть якщо зловмисник зможе обійти правила 3 та 4 рівнів, він все ще може бути обмежений мережевими політиками 7 рівня. Додатково, Cilium має такий функціонал як «cluster mesh» з вбудованим mTLS, який може легко інтегруватися з інструментами безпеки середовища виконання контейнерів (такими як Tetragon) для виявлення підозрілих системних викликів за допомогою eBPF. Узагальнюючи, дизайн даного мережевого плагіну надає високий рівень видимості та контролю, що дозволяє

адміністраторам аналізувати потоки даних за допомогою таких інструментів як Hubble і реагувати на загрози в режимі реального часу.

Flannel

Це старий та простий CNI створений компанією CoreOS, яка зараз є частиною Red Hat. Його дизайн дуже прямолінійний та передбачає, як основне завдання, призначення підмереж кожній ноді та енкапсуляцію трафіку між ними [9]. Він підтримує різні рішення для енкапсуляції, але основним є VXLAN, який створює оверлейну мережу. З цим рішенням кожна нода має свій тунельний інтерфейс і Flannel енкапсулює пакети між подами у UDP пакети та надсилає їх мережею кластера. На кожній ноді Flannel налаштовує мережевий міст, куди додаються всі інтерфейси под, а інтерфейсу моста присвоюється IP адреса з підмережі нод. Тобто, він створює оверлей 3 рівня, де IP адреси под видимі тільки для Flannel, який співвідносить їх до IP адрес справжньої мережі.

Щодо безпеки цього рішення, його простота є основною перевагою, так як є менше компонентів, які взаємодіють між собою. Для прикладу, він не використовує складного демона, який запущений на кожній ноді, а обмежується невеличким агентом для розподілення адрес підмереж через etcd чи Kubernetes API. І, зрештою, чим менший код – тим менше потенційних проблем у ньому. Повертаючись до оверлеїв (у режимі VXLAN), поди на нодах розміщені у різних доменах 2 рівня, тому навіть поди на одній ноді повинні комунікувати через мережевий міст. З цієї причини зловмисник, який зміг скомпроментувати поду, буде змушений також скомпроментувати VXLAN, щоб перехоплювати чужий трафік. Цікавим є також те, що зловмисник з привілейованим доступом на одній ноді може спробувати прослуховувати трафік, але без ключа VXLAN та комплексного підходу до цієї задачі це буде дуже важко зробити. І нарешті, хотілося б нагадати, що Flannel не використовує фільтрацію пакетів та їх маршрутизацію на рівні ядра, а це автоматично виключає вплив відповідних вразливостей на роботу плагіну.

Однак, простота дизайну має також і свої мінуси, так як плагін не було розроблено з врахуванням постійно зростаючих вимог щодо безпеки. З цього випливає важлива особливість – відсутність підтримки мережевих політик Kubernetes. Іншими словами, комунікація між подами абсолютно не обмежена. Ця загроза є особливо актуальною у кластерах з багатьма користувачами або з вимогою нульової довіри до мережі. Але, користувачі які використовують Flannel, часто розгортають його з окремим контролером мережевих політик (для прикладу, Canal, який об’єднує Flannel для конфігурації мережі і Calico для впровадження мережевих політик).

Іншим недоліком даного плагіну є відсутність шифрування трафіку під час його енкапсуляції. Якщо зловмисник має доступ до основної мережі (для прикладу, скомпроментувавши роутер), він може перехоплювати весь мережевий трафік всередині кластера. На жаль, Flannel не має вбудованої опції шифрування трафіку, як Calico та Cilium, тому потрібно налаштовувати необхідні рішення на рівні хостів (для прикладу, IPsec).

Weave Net

Цей плагін розроблений компанією Weaveworks, який створює віртуальну меш-мережу між усіма нодами [10]. Він може функціонувати у двох режимах: «швидкий шлях даних» (використовуючи модулі ядра та швидке перенаправлення простору користувача) або «резервний режим простору користувача». Плагін формує оверлейну мережу та енкапсулює пакети, використовуючи власний UDP протокол. Цікавою особливістю даного рішення є фокус на простоту у використанні. Він також може шифрувати трафік, використовуючи ключ клієнта, та автоматично виявляє топологію мережі. Цей плагін, як і Flannel, створює віртуальний мережевий міст на кожній ноді та приєднує контейнери до нього, але використовуючи зовсім інший підхід. Всі поди формують меш-систему і знають про існування один одного, тому можуть надсилати пакети через оверлей до правильного вузла, де розташована необхідна пода. На жаль, компанія, яка розробила цей плагін, оголосила про закриття бізнесу у лютому 2024 року, але аналіз його безпеки та архітектури важливий з наукової точки зору.

Фокусуючись на безпеці даного рішення, Weave Net був одним із перших CNI, які пропонували вбудоване шифрування. У разі увімкнення цього функціоналу, використовується секретний ключ, який шифрує весь оверлейний трафік (за допомогою швидких шифрів у просторі ядра), надаючи захист від сніффінгу. Також плагін підтримує мережеві політики Kubernetes, що дозволяє обмежувати комунікацію між подами, як у Calico. Під капотом Weave Net використовує проксі простору імен Weave NPC (Network Policy Controller), яка керує застосуванням політик. Іншою перевагою плагіну є використання мешу. Це означає, що система не має єдиної точки відмови для маршрутизації мережевого трафіку. Навіть якщо одна нода була скомпроментована, це не надає зловмиснику повний доступ до маршрутизації всієї мережі (на відміну від Calico та інших рішень із централізованим контролером).

Підсумок відмінностей у забезпеченні безпеки плагінів

Flannel – простий плагін, який потребує зовнішнього інструменту для сегментації мережі. Calico та Cilium - впроваджують мережеві політики і підтримують шифрування трафіку, але використання Calico зовнішніх протоколів (BGP) додає ризик для панелі керування (control plane), тоді як залежність Cilium від eBPF додає ризик на рівні ядра. Weave Net забезпечує оверлейну мережу з можливістю шифрування. Всі плагіни запускають привілейовану поду на кожній ноді, що в свою чергу створює ризик компроментації цієї поди або логіки CNI. Такий сценарій призведе до надання зловмиснику міцної точки опори, так як він буде мати привілейований доступ до ноди і можливість маніпулювання трафіком. Саме тому, безпека мережі у Kubernetes рівноцінна безпеці плагінів CNI та їх відповідних конфігурацій. Відповідні порівняння структуровано в наступній Табл.1.

Таблиця 1

Порівняння функцій безпеки плагінів CNI

CNI плагін	**Підтримка мережевих політик**	**Підтримка шифрування**	**Рівень привілеїв необхідних для функціонування**
Cilium	Так; підтримує політики 7 рівня	Так; З допомогою IPsec або WireGuard	Привілейована пода на кожній ноді з повним доступом до ядра. Використовує повноваження, такі як NET_ADMIN, CAP_BPF
Calico	Так; підтримує політики 7 рівня	Так; з допомогою IPsec або WireGuard	Привілейована пода на кожній ноді. Залежно від режиму роботи, потребує або невеликий набір повноважень для конфігурації правил iptables, або повноваження схожі до Cilium
Flannel	Ні	Ні; можливість шифрування трафіку через IPsec колись існувала, але без офіційної підтримки	Привілейована пода на кожній ноді. Використовує повноваження NET_ADMIN, NET_RAW для створення VXLAN інтерфейсів та маршрутів
Weave Net	Так	Так; з допомогою клієнтського ключа (AES)	Привілейована пода на кожній ноді. Має доступ до мережі нод, що необхідно для створення мешу, та використовує повноваження NET_ADMIN.

Результати досліджень та висновки

Дослідження вказує на те, що більшість мережевих плагінів Kubernetes впроваджують різноманітні засоби захисту як на рівні дизайну та архітектури програмного рішення, так і на рівні налаштувань самого плагіна. Однак, є багато шляхів для покращення їх безпеки. Саме тому, підсумовуємо ключові результати дослідження та робимо висновки:

- *Сегментація мережі є надзвичайно важливою, але її реалізація відрізняється*. CNI плагіни впроваджують сегментацію через мережеві політики. Calico та Cilium надають надійні механізми впровадження політик як частину функціоналу плагіну, тоді як Flannel потребує об'єднання з іншою системою для виконання цього завдання. Weave Net займає позицію посередині, підтримуючи мережеві політики, але виконуючи це завдання з високими привілеями, що створює певні ризики. З теоретичної точки зору, найбезпечнішим є сценарій, коли кожна пода «бачить» лише ті кінцеві точки мережі, які має (принцип найменших привілеїв). Одне із досліджень продемонструвало підхід до досягнення цієї цілі за допомогою виділених мережевих пісочниць для кожного контейнера [11]. Хоч це й не було впроваджено у CNI специфікації, однак цей концепт вплинув на розвиток такого функціоналу Cilium як доступ, що базується на ідентифікації. Підсумовуючи, використання мережевих політик є базовою вимогою для безпечних кластерів Kubernetes. Аналіз відкритих джерел підтверджує, що кластери без цих політик є дуже вразливим до подальшого переміщення зловмисника всередині мережі у разі злому будь-якого поду, що також підтверджується у інших дослідженнях.

- *Площини обробки даних, що базуються на eBPF, проявляють високі перспективи (але водночас несуть нові ризики)*. eBPF підхід Cilium та Calico показує вищу продуктивність системи та кращі результати у забезпеченні безпеки мережі. Можливість застосовувати мережеві політики на рівні ядра і навіть фільтрувати метадані 7 рівня (наприклад, HTTP методи) можуть запобігти певним сценаріям атак (використання нестандартних портів чи протоколів для викрадення даних). Крім того, використання даної технології дозволяє побудувати низькорівневу систему моніторингу (на основі сигналів на рівні ядра), що дозволяє значно покращити захист всієї системи [5]. З іншого боку, залежність від ядра має свої ризики. Коли знаходять вразливості у ядрі, такі як CVE-2021-31440, будь-яка система, що залежить від вразливого функціоналу ядра, стає ціллю зловмисника, поки ядро системи не оновлено до новішої версії. Узагальнюючи, використання eBPF є надійним способом забезпечення безпеки мережі, але потрібно якомога частіше оновлювати ядро системи та зміцнювати його різноманітними способами, такими як BPM LSM або вводити обмеження виконання непривілейованих BPF, що захистить систему від невідомих вразливостей.

- *Важливість принципу найменших привілеїв та зміцнення захисту системи*. Багато вразливостей не несли б в собі ризики, якби було дотримано принципів найменших привілеїв. Для прикладу, проблема Weave Net з hostPID [12] мала значення тільки тому, що контейнер міг бачити всі процеси на хості, хоча йому цього й не було потрібно. Саме тому, зміцнення захисту ноди та подів CNI є необхідними для забезпечення безпеки мережі. Це включає такі кроки як запуск подів CNI тільки з найменш необхідними дозволами та використання SELinux для обмеження можливостей цих подів.

- *Впровадження найкращих практик*

  - *Моніторинг та аудит*. Очевидною є тенденція покращення видимості щодо мережевого трафіку. Для прикладу, інструмент Hubble від Cilium, а також функціонал Weave Net для відстежування метрик з'єднань, що допомогає виявляти різноманітні атаки (для прикладу, сканування портів). Зберігання та аналіз мережевих логів також є необхідною умовою для покращення стану безпеки мережі, що дозволяє виявляти атиповий трафік з под та швидко реагувати на спроби просування по мережі зі скомпроментованої поди.

- *Регулярне оновлення плагінів*. Просте правило, яке застосовне не тільки до розглянутої теми, але й впровадження якого значно покращить безпеку мережі Kubernetes, так як новіші версії плагінів застосовують оновлений код, який не вразливий до новіших CVE.
- *Підхід «Захисту у глибину»*. Застосування мережевих політик є хорошим способом забезпечення захисту мережі, але впровадження комплексного підходу до вирішення цієї задачі є кращим рішенням. Написання політик безпеки для подів, щоб уникнути виконання їх у привілейованому режимі, використання інструментів безпеки, таки як Falco, для ідентифікації незвичайної поведінки всередині мережевих под, і навіть зміцнення захисту ядра – усе це способи, які покращують безпеку мережевого середовища.

- *Загальний стан безпеки плагінів*. Як уже згадувалось, плагіни застосовують різноманітні засоби, заходи та методи забезпечення своєї безпеки, але потребують правильного налаштування та регулярного обслуговування. Кожен з них має свій «профіль поверхні атаки»:
  - Calico: має хороші способи застосування політик, але потребує контролю протоколів «площини керування» (для прикладу, забезпечення захисту BGP)
  - Cilium: дуже надійне та перспективне рішення, але потребує регулярних оновлень ядра та обмеження доступу до API/сокетів керування плагіном
  - Flannel: простий та стабільний, але нестачу функціоналу необхідно компенсувати додатковими механізмами безпеки
  - Weave Net: зручний завдяки вбудованому шифруванню, але потрібно запускати його у непривілейованому режимі та зважати на взаємодію хоста та оверлею

Отож, узагальнюючи, з теоретичної точки зору, майбутня робота щодо покращення безпеки мережевих плагінів може включати ідеї, схожі на ті, що пропонує Bastion [11], або інші академічні напрацювання. Для прикладу, забезпечення мережевих стеків для кожного орендаря (tenant) або більш глибоку ізоляцію мережевих ресурсів. Зважаючи на стрімку еволюцію загроз, CNCF та Kubernetes SIG Network і далі вважають безпеку одним із пріоритетних завдань, і очікується, що наступне покоління CNI ще більше знизить ризик мережевих атак у Kubernetes.

Перелік посилань

1. With Kubernetes, the U.S. Department of Defense is enabling DevSecOps on F-16s and battleships (2020). CNCF. https://www.cncf.io/case-studies/dod/
2. CNCF SURVEY 2019 (2019). CNCF. https://www.cncf.io/wp-content/uploads/2020/08/CNCF_Survey_Report.pdf
3. Minna F. et al. (2021). Understanding the Security Implications of Kubernetes Networking. IEEE Security & Privacy, 19(5), 46–54. https://balakrishnanc.github.io/papers/minna-ieeesp2021.pdf
4. Budigiri G. et al. (2021). Network Policies in Kubernetes: Performance Evaluation and Security Analysis. Joint EuCNC & 6G Summit. https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482526
5. Yevle D. (2024). Exploring eBPF and its Integration with Kubernetes. OpenSourceForU. https://www.opensourceforu.com/2024/12/exploring-ebpf-and-its-integration-with-kubernetes/
6. Calico Official Documentation. https://docs.tigera.io/calico/latest/about/
7. Hoffman K. (2023). Comparing Networking Solutions for Kubernetes: Cilium vs. Calico vs. Flannel. Civo Blog. https://www.civo.com/blog/calico-vs-flannel-vs-cilium
8. Cilium Official Documentation. https://docs.cilium.io/en/stable/
9. Flannel Official Documentation. https://github.com/flannel-io/flannel
10. Weave Net Official Documentation. https://github.com/weaveworks/weave/tree/master
11. Nam J. et al. (2020). BASTION: A Security Enforcement Network Stack for Container Networks. У 2020 USENIX Annual Technical Conference. http://www.usenix.org/system/files/atc20-nam.pdf
12. Weaveworks (2020). Weave Net 2.8.0 Release Notes (Removal of hostPID and other hardening). OpenCVE. https://app.opencve.io/cve/?vendor=weave

Надійшла 28.02.2025