



DOI 10.28925/2663-4023.2026.35.1438

УДК 004.056:629.331

Журавчак Анастасія Юрїївна

доктор філософії, асистент кафедри захисту інформації

Національний університет «Львівська політехніка», Львів, Україна

ORCID: 0000-0002-8196-7963

anastasii.a.y.tolkachova@lpnu.ua

Журавчак Даниїл Юрїйович

доктор філософії, доцент кафедри захисту інформації

Національний університет «Львівська політехніка», Львів, Україна

ORCID: 0000-0003-4989-0203

danyil.y.zhuravchak@lpnu.ua

Журавчак Юрїй Юрїйович

аспірант, асистент кафедри захисту інформації

Національний університет «Львівська політехніка», Львів, Україна

ORCID: 0009-0003-2378-5365

yurii.zhuravchak.asp.2026@lpnu.ua

ВИКОРИСТАННЯ eBPF ДЛІЯ ЗАХИСТУ АВТОМОБІЛЬНИХ КОНТЕЙНЕРИЗОВАНИХ СЕРЕДОВИЩ

Анотація. Програмно-визначені автомобілі дедалі частіше використовують Linux-платформи, централізовані обчислювальні вузли та контейнеризовані застосунки. Спільне ядро операційної системи створює ризики порушення ізоляції, підвищення привілеїв і несанкціонованого доступу контейнерів до ресурсів та внутрішніх мереж транспортного засобу. Метою роботи є систематизація підходів до застосування технології eBPF для моніторингу, виявлення загроз і реалізації політик безпеки в автомобільних контейнеризованих середовищах під час виконання. Методологічну основу становлять аналіз наукових публікацій і технічної документації ядра Linux та порівняння механізмів захисту контейнерів і автомобільних мереж. Систематизовано функції eBPF (моніторинг, виявлення, контроль доступу, контроль SocketCAN, кореляція, реагування) та визначено основне обмеження кожної з них в автомобільному середовищі. Обґрунтовано розмежування реєстрації подій і синхронного блокування операцій засобами BPF LSM; дворівневий контроль взаємодії з SocketCAN, що окремо перевіряє право використання CAN-інтерфейсу та відповідність передавання призначенню сервісу; формування профілів поведінки для окремих режимів роботи платформи; явне розрізнення встановленого, імовірного та невідомого джерела подій під час кореляції активності контейнера з CAN-трафіком. Запропоновано архітектуру з трьох компонентів (засоби ядра, локальний аналітичний агент, компонент керування політиками) та критерії оцінювання: повноту збирання подій, частку втрат телеметрії, кількість хибних тривог за годину роботи, кількість необґрунтованих блокувань і вплив на часові характеристики прикладних завдань. Практичне значення роботи полягає у визначенні вимог до засобів захисту та порядку їх поетапної перевірки на автомобільних Linux-платформах. Перспективи досліджень пов'язані з експериментальною перевіркою запропонованих рішень на апаратних платформах автомобільного класу.

Ключові слова: eBPF; кібербезпека; контейнеризація; Linux; моніторинг; виявлення втротгень; SocketCAN; CAN-шина; BPF LSM; програмно-визначений автомобіль.

ВСТУП

Сучасні автомобілі поступово трансформуються з набору відносно ізольованих електронних блоків керування у складні програмно-визначені кіберфізичні системи. У таких системах дедалі ширше застосовуються Linux-платформи, централізовані обчислювальні вузли та контейнеризація, що забезпечують ізоляцію програмних компонентів, спрощують розгортання оновлень і дають змогу динамічно розширювати функціональність транспортного засобу. Водночас збільшення обсягу



програмного забезпечення, підключення автомобіля до зовнішніх мереж і спільне використання апаратних ресурсів різними застосунками розширюють поверхню атак [1]. Компрометація контейнера або привілейованого процесу може призвести не лише до порушення конфіденційності даних, а й до несанкціонованого доступу до системних пристроїв, мережеских інтерфейсів та внутрішньої автомобільної мережі.

Одним із перспективних засобів контролю безпеки Linux-систем є технологія extended Berkeley Packet Filter, або eBPF. Вона дає змогу без внесення змін до вихідного коду ядра виконувати перевірені верифікатором програми у просторі ядра та отримувати детальну інформацію про системні виклики, процеси, файлові операції, мережескі з'єднання і взаємодію контейнерів із ресурсами операційної системи. Поєднання механізмів спостереження, фільтрації та застосування політик робить eBPF потенційною основою для побудови засобів захисту автомобільних контейнеризованих середовищ під час виконання. Зокрема, eBPF може застосовуватися для виявлення нетипової поведінки процесів, контролю доступу до пристроїв, обмеження мережескої активності, моніторингу взаємодії із SocketCAN та локального реагування на ознаки компрометації.

Однак перенесення наявних eBPF-підходів із хмарних і серверних середовищ до автомобільних систем потребує врахування обмежених обчислювальних ресурсів, вимог до передбачуваної затримки, функціональної безпеки та змішаного рівня критичності програмних компонентів. Саме ці відмінності визначають, які з наявних рішень можна перенести до автомобільних платформ без суттєвих змін, а які потребують адаптації.

Метою статті є систематизація підходів до використання технології eBPF для моніторингу, виявлення загроз і захисту автомобільних контейнеризованих середовищ під час виконання. Особливу увагу приділено визначенню можливостей та обмежень цих підходів з урахуванням доступних обчислювальних ресурсів, вимог до затримки й функціональної безпеки автомобільних систем.

Постановка проблеми. Поширення Linux-платформ і контейнеризації в сучасних автомобілях розширює поверхню атак та створює ризики несанкціонованого доступу до системних ресурсів і внутрішніх мереж транспортного засобу. Традиційні засоби захисту контейнерів не повною мірою враховують вимоги автомобільних систем до обмежених ресурсів, передбачуваної затримки та функціональної безпеки. Технологія eBPF дає змогу здійснювати низькорівневий моніторинг і контроль активності процесів та контейнерів, однак наявні підходи до її використання в автомобільному середовищі залишаються фрагментарними. Тому актуальною є систематизація можливостей, обмежень і перспектив застосування eBPF для захисту автомобільних контейнеризованих середовищ під час виконання.

Аналіз останніх досліджень і публікацій. У сучасних дослідженнях захист контейнеризованих середовищ дедалі тісніше пов'язується з контролем поведінки застосунків під час виконання. Зокрема, Gwak та співавтори [2] запропонували систему KRSIE, яка використовує механізми BPF LSM для застосування політик безпеки до контейнерів Kubernetes. Ця робота демонструє можливості eBPF не лише для спостереження за активністю контейнерів, а й для обмеження небажаних операцій. Близький підхід реалізовано в BPFContain [3], де eBPF-програми та механізм KRSI (BPF LSM) застосовано для обмеження контейнерів відповідно до заданих політик. Jarkas та співавтори [4] систематизували вразливості, атаки й засоби захисту контейнерів, приділивши увагу ризикам спільного використання ядра та порушення ізоляції. Методологічний підхід до вибору засобів моніторингу представлено в роботі Jolak та співавторів [5]. Запропонований ними фреймворк CONSERVE дає змогу зіставляти методи моніторингу за об'єктами спостереження, стратегіями виявлення та умовами застосування. Автомобільну специфіку захисту розкрито в огляді Kifor і Popescu [6], у якому розглянуто стандарти, підходи до тестування та технології моніторингу кібербезпеки транспортних засобів. Ця праця формує контекст для оцінювання засобів захисту з урахуванням особливостей автомобільної архітектури. Приклад використання eBPF для виявлення шкідливої активності наведено в дослідженні Kim та співавторів [7]. Автори поєднали збирання подій за допомогою Tetragon із методами машинного навчання для виявлення криптоджекінгу в контейнерах. Водночас отримані результати стосуються хмарного середовища, тому їх перенесення до автомобільних платформ потребує окремого оцінювання ресурсних витрат і затримок. Значну увагу приділено також виявленню вторгнень в автомобільні мережах. Lampe і Meng [8] узагальнили загрози та підходи до виявлення атак у різних сегментах автомобільної комунікаційної інфраструктури. В іншій роботі цих авторів [9] систематизовано методи глибокого навчання для автомобільних систем виявлення вторгнень, розглянуто набори даних, типи атак і показники оцінювання. Для теми цієї статті зазначені огляди важливі як основа вибору методів аналізу подій, однак самі по собі вони не визначають способу встановлення зв'язку між мережевою аномалією та процесом усередині контейнера.

Окремий напрям досліджень охоплює аудит і забезпечення ізоляції контейнерів на рівні операційної системи. Lim та співавтори [10] запропонували saBPF — розширення eBPF для системного аудиту з деталізацією до окремого контейнера. Такий підхід створює основу для аналізу походження подій і



застосування індивідуальних політик аудиту. Lin та співавтори [11] дослідили безпеку Linux-контейнерів, розглянувши атаки та відповідні контрзаходи. У контексті автомобільної платформи ці роботи дають підстави розглядати захист контейнера разом із контролем його доступу до ресурсів спільного ядра.

Проблематику захисту внутрішньої автомобільної мережі розглянуто в огляді Lokman та співавторів [12], присвяченому вразливостям CAN-шини та методам виявлення вторгнень. Натомість Nam та співавтори [13] дослідили використання XDP/eBPF для захисту міжконтейнерних комунікацій. Ці напрями охоплюють різні рівні взаємодії: автомобільну мережу та мережевий обмін між контейнерами. Їх поєднання потребує врахування особливостей SocketCAN і не передбачає автоматичного перенесення механізмів оброблення IP-трафіку на CAN-повідомлення. Архітектурні передумови застосування контейнерів в автомобілях представлено в роботі Pan та співавторів [14], яка поєднує модельно-орієнтоване проєктування, розподіл ресурсів і розгортання програмного забезпечення із застосуванням віртуалізації. Rajapaksha та співавтори [15] систематизували методи штучного інтелекту для виявлення вторгнень у внутрішніх автомобільних мережах, включно з ознаками, наборами даних та обмеженнями наявних підходів. Shin та співавтори [16] розглянули поєднання eBPF та AdaBoost для виявлення атак на контейнери. Огляд Young та співавторів [17] доповнює ці дослідження систематизацією методів виявлення атак у CAN-мережах, їхніх переваг та обмежень.

Аналіз наведених публікацій свідчить, що захист контейнерів засобами eBPF, віртуалізація автомобільних платформ і виявлення атак у CAN-мережах переважно розглядаються як окремі напрями. У розглянутих працях недостатньо висвітлено їх спільне застосування для зіставлення активності контейнера з подіями автомобільної мережі за одночасного врахування ресурсних обмежень, передбачуваної затримки та функціональної безпеки. Тому внесок цієї статті полягає в систематизації відповідних підходів за функціями моніторингу, виявлення, контролю доступу, контролю взаємодії з SocketCAN, кореляції та реагування, а також у визначенні умов і критеріїв їх оцінювання для автомобільних контейнеризованих середовищ.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результати дослідження викладено за вісьмома взаємопов'язаними аспектами: модель загроз і вимоги автомобільної платформи; механізми eBPF, придатні для захисту; порівняння наявних рішень; функціональна систематизація застосування eBPF; контроль взаємодії контейнерів із SocketCAN; кореляція активності контейнерів із подіями автомобільної мережі; узагальнена архітектура і порядок реагування; критерії та порядок оцінювання.

1. Модель загроз і вимоги автомобільної платформи.

Експериментальний аналіз поверхні атак автомобіля [1] показав, що компрометація можлива через опосередкований фізичний доступ (діагностичний порт, мультимедійні носії), бездротові канали ближнього радіуса дії та канали дальнього радіуса дії, зокрема телематичний зв'язок через стільникові мережі, а доступ, отриманий через один електронний блок керування, дає змогу впливати на внутрішню мережу транспортного засобу. На централізованій Linux-платформі ці вектори сходяться на одному обчислювальному вузлі: телематичні, мультимедійні та діагностичні сервіси виконуються в окремих контейнерах, але спільно використовують ядро операційної системи. Систематизації загроз для контейнерів [4], [11] вказують на основні класи ризиків такої конфігурації: вихід за межі контейнера через вразливості ядра, надмірні привілеї та можливості процесів, помилки конфігурації монтування й доступу до пристроїв, а також виснаження спільних ресурсів.

Для автомобільної платформи до цих загроз додаються специфічні наслідки: несанкціоноване передавання CAN-повідомлень, зловживання діагностичними функціями та порушення часових характеристик функцій, пов'язаних із безпекою руху. Тому модель загроз доцільно формувати в межах процесу аналізу загроз і оцінювання ризиків, передбаченого ISO/SAE 21434 [18], а засоби захисту проєктувати так, щоб вони не порушували вимог функціональної безпеки ISO 26262 [19] до функцій різного рівня цілісності (ASIL), розміщених на одній платформі. Узагальнення класів загроз і відповідних точок спостереження та контролю eBPF наведено в таблиці 1.

Таблиця 1

Класи загроз автомобільної контейнеризованої платформи та точки спостереження й контролю eBPF

Клас загрози	Ознаки прояву	Точки спостереження й контролю eBPF
Вихід за межі контейнера, підвищення	Зміна облікових даних процесу, виклики mount, setns, unshare,	Точки трасування системних викликів; LSM-хуки cred_prepare,



Клас загрози	Ознаки прояву	Точки спостереження й контролю eBPF
привілеїв	спроби завантаження модулів ядра	task_fix_setuid, sb_mount, kernel_load_data, kernel_read_file
Виконання стороннього коду	Запуск неочікуваних виконуваних файлів, відображення в пам'ять бібліотек із тимчасових каталогів	Точка трасування sched_process_exec; LSM-хуки bprm_check_security, mmap_file
Несанкціонований доступ до CAN-шини	Створення сокетів AF_CAN, прив'язування до CAN-інтерфейсу, передавання кадрів	LSM-хуки socket_create, socket_bind, socket_sendmsg
Витік даних і віддалене керування	Неочікувані зовнішні з'єднання телематичних і мультимедійних сервісів	LSM-хук socket_connect; cgroup-програми connect4/connect6; програми TC
Виснаження ресурсів	Аномальна інтенсивність системних викликів, мережевого чи CAN-обміну	Лічильники в картах eBPF; обмеження ресурсів — засобами cgroup поза eBPF
Атаки на засоби захисту	Завантаження чи від'єднання eBPF-програм, зміна карт політик	LSM-хук bpf; мінімізація можливостей CAP_BPF і CAP_SYS_ADMIN

Зазначені точки контролю є необхідною, але не достатньою умовою захисту: їх наявність залежить від версії та конфігурації ядра, а обґрунтованість рішень — від якості політик і контексту, доступного в момент виклику.

2. Механізми eBPF та їх придатність для автомобільної платформи.

Технологічною передумовою застосування eBPF є перевірка завантажуваних програм верифікатором ядра, який аналізує допустимість доступу до пам'яті, стан реєстрів, використання вказівників, завершуваність програми та відповідність аргументів допоміжних функцій установленим обмеженням [20]. Однак успішна верифікація не засвідчує правильності політики безпеки і не гарантує виконання часових вимог автомобільної функції: програма може бути прийнятною для верифікатора, але містити надто широке правило доступу або створювати надмірне навантаження за високої частоти подій. Тому перевірку eBPF-програми, перевірку змісту політики та оцінювання її впливу на платформу слід розглядати як окремі завдання.

Принциповим для побудови захисту є розмежування реєстрації події та примусового застосування правила. Програми трасування (точки трасування, kprobes, fentry/fexit) повідомляють про операцію, але не забороняють її: до моменту оброблення події в просторі користувача операція вже може завершитися. Механізм BPF LSM дає змогу приєднувати програми до хуків модуля безпеки Linux і для хуків із відповідною семантикою повертати код помилки, тобто синхронно забороняти операцію [21]. Його використання потребує ядра, зібраного з параметром CONFIG_BPF_LSM, та наявності bpf у списку активних LSM, тому доступність механізму необхідно перевіряти для конкретної цільової платформи. Це суттєво для автомобільних систем, які часто використовують ядра постачальників апаратних платформ із власною конфігурацією; від наявності в таких ядрах інформації про типи (BTF) залежить і можливість переносимого розгортання eBPF-програм без їх перекомпіляції для кожної версії ядра.

Окрему групу становлять програми, що приєднуються до контрольних груп (cgroup): вони природно відповідають межах контейнера і дають змогу контролювати доступ до файлів пристроїв, створення мережесокетів і встановлення з'єднань. Водночас їх охоплення обмежене: програми контролю пристроїв діють лише щодо пристроїв, представлених файлами, а програми, які спрацьовують під час створення сокета, орієнтовані на сокети IPv4 та IPv6. Для мережевого рівня застосовуються програми XDP і TC, придатність яких для захисту міжконтейнерного обміну показано в [13]; проте ці механізми розраховані на IP-трафік, і їх перенесення на CAN-інтерфейси потребує окремої перевірки.

Передавання подій до простору користувача зазвичай здійснюється через кільцевий буфер eBPF, який спільно використовується кількома процесорами і зберігає порядок записаних подій [22]. Буфер має скінченну місткість, тому спроба резервування місця може бути невдалою. Для автомобільного застосування необхідно вести лічильники втрат, установлювати пріоритети подій і скорочувати деталізацію другорядних спостережень під час перевантаження; відсутність повідомлень за наявності втрат не може інтерпретуватися як відсутність небажаної активності. Узагальнену характеристику механізмів наведено в таблиці 2.



Таблиця 2

Механізми eBPF і їх придатність для захисту автомобільних контейнеризованих середовищ

Механізм	Призначення	Синхронна заборона	Обмеження в автомобільному середовищі
Трасування (tracepoints, kprobes, fentry/fexit)	Спостереження за системними викликами та функціями ядра	Ні	kprobes прив'язані до внутрішніх функцій ядра, які можуть змінюватися між версіями
BPF LSM	Контроль операцій у точках перевірки безпеки	Так	Потребує CONFIG_BPF_LSM і активації bpf серед LSM; помилка політики призводить до відмови легітимної операції
cgroup-програми	Контроль пристроїв, сокетів і з'єднань у межах контейнера	Так, для свого класу операцій	Не охоплюють CAN-інтерфейси як мережеві пристрої та сокети AF_CAN
XDP, TC	Фільтрація мережевого трафіку на інтерфейсі	Так (відкидання пакетів)	Орієнтовані на IP-трафік; застосовність до CAN потребує перевірки
Карти та кільцевий буфер	Зберігання політик і стану, передавання подій	—	Скінченна місткість, ризик втрат телеметрії під навантаженням

3. Порівняльний аналіз наявних рішень.

Для оцінювання придатності наявних рішень до автомобільних платформ розглянуті праці зіставлено за використанням механізмом, функцією захисту, середовищем, для якого їх розроблено, та аспектами, що потребують адаптації (табл. 3).

Таблиця 3

Порівняння підходів до захисту контейнерів і автомобільних мереж

Робота	Механізм	Функція захисту	Середовище	Що потребує адаптації
KRSIE [2]	BPF LSM, інструментування	Застосування політик безпеки до контейнерів	Kubernetes	Ресурсні обмеження; сокети AF_CAN не розглянуто
BPFContain [3]	eBPF, KRSI (BPF LSM)	Обмеження контейнерів за політиками	Linux-хост із контейнерами	Режимні політики; контроль CAN-взаємодії
Kim та ін. [7]	Tetragon, машинне навчання	Виявлення криптоджекінгу	Хмарне середовище	Витрати на аналіз на вузлі; інші класи загроз
saBPF [10]	Розширення eBPF для аудиту	Аудит із деталізацією до контейнера	Linux-хост із контейнерами	Сумісність із ядрами автомобільних платформ
Nam та ін. [13]	XDP/eBPF	Захист міжконтейнерного мережевого обміну	Контейнерні IP-мережі	CAN-трафік не охоплено
Shin та ін. [16]	eBPF, AdaBoost	Виявлення атак на контейнери	Linux-хост із контейнерами	Витрати на класифікацію; зміна поведінки між режимами
CIDS [23]	Часові характеристики CAN-трафіку (без eBPF)	Виявлення підміни електронних блоків керування	Автомобільна CAN-мережа	Відсутній зв'язок із процесами Linux-вузла
SafeBPF [24]	Програмна та апаратна (ARM MTE) ізоляція	Захист середовища виконання eBPF	Платформи ARM	Доступність MTE на автомобільних SoC



Зіставлення дає змогу зробити чотири узагальнення. По-перше, розглянуті eBPF-рішення для захисту контейнерів [2], [3], [7], [10], [13], [16] розроблено й оцінено для серверних або хмарних середовищ, де обчислювальні ресурси, вимоги до затримки та наслідки хибного блокування суттєво відрізняються від автомобільних. По-друге, жодне з них не розглядає контроль сокетів AF_CAN і взаємодії контейнерів із CAN-інтерфейсами. По-третє, системи виявлення вторгнень у CAN-мережах [12], [15], [17], [23] аналізують трафік шини без контексту процесів і контейнерів обчислювального вузла, тоді як засоби захисту контейнерів не використовують ознак автомобільної мережі. По-четверте, якість методів виявлення переважно оцінюється за показниками класифікації на наборах даних, а не за наслідками для функцій транспортного засобу. Ці розриви визначають зміст подальшої систематизації.

4. Функціональна систематизація застосування eBPF.

За результатами аналізу виокремлено шість функціональних напрямів застосування eBPF в автомобільних контейнеризованих середовищах (табл. 4). Для кожного напрямку визначено основне завдання, базові механізми та головне обмеження, яке необхідно враховувати під час перенесення рішень із хмарних середовищ.

Таблиця 4

Функціональні напрями застосування eBPF та обмеження їх автомобільної адаптації

Напрямок	Завдання в автомобільному середовищі	Базові механізми	Основне обмеження
Моніторинг	Пов'язування системних подій із процесами та контейнерами	Трасування, кільцевий буфер	Неповний контекст і можливі втрати телеметрії
Виявлення	Зіставлення подій із правилами та режимними профілями	Карти профілів, аналіз у просторі користувача	Хибні спрацювання під час легітимних змін поведінки
Контроль доступу	Заборона окремих операцій у придатних точках перевірки	BPF LSM, sgroup-програми	Залежність від семантики хуків і конфігурації ядра
Контроль SocketCAN	Перевірка дозволеної взаємодії з CAN-інтерфейсами	LSM-хуки сокетних операцій	Необхідність охоплення різних шляхів передавання
Кореляція	Поєднання локальної активності з мережевими ознаками	Ідентифікатори sgroup, часові позначки	Відсутність повної атрибуції зовнішніх повідомлень
Реагування	Обмеження небажаних дій зі збереженням потрібних функцій	BPF LSM, оновлення карт політик	Можливий вплив на доступність автомобільних сервісів

Моніторинг доцільно організовувати навколо подій, значущих для обраної моделі загроз, добираючи мінімально достатній набір спостережень для кожного класу застосунків. Для телематичного сервісу це запуск нових виконуваних файлів, зміна облікових даних, звернення до конфігураційних ресурсів і встановлення зовнішніх зв'язків; для діагностичного компонента додатково важливі відкриття CAN-інтерфейсів і зміна параметрів їх використання. Збирання максимальної кількості подій без визначеної аналітичної мети ускладнює інтерпретацію даних і збільшує витрати ресурсів. Запис події має містити часову позначку, тип операції, її результат, ідентифікатор процесу та ознаки належності до контейнера, зокрема ідентифікатор контрольної групи. Окремої уваги потребує життєвий цикл ідентифікаторів: після завершення процесу чи перезапуску контейнера попередній зв'язок не повинен переноситися на новий об'єкт, тому таблиці відповідності слід оновлювати разом зі зміною складу сервісів, а невизначений контекст позначати явно.

Виявлення загроз доцільно будувати як поєднання явних правил і поведінкових профілів. Явні правила визначають операції, несумісні з призначенням контейнера, наприклад спробу мультимедійного застосунку створити CAN-сокет. Поведінковий профіль описує очікувані послідовності та інтенсивність подій; одиничне відхилення є підставою для додаткової перевірки, тоді як поєднання кількох пов'язаних ознак підвищує обґрунтованість рішення про інцидент. Особливістю автомобільної платформи є залежність нормальної поведінки від режиму роботи: завантаження системи, штатний рух, технічне обслуговування та оновлення програмного забезпечення передбачають різні набори дозволених операцій. Тому доцільно формувати окремі профілі для кожного режиму і керувати переходами між ними через довірений компонент; сам контейнер не повинен мати змоги оголосити перехід у режим із розширеними повноваженнями. Автоматично сформовані профілі потребують перевірки, оскільки небажана активність у період навчання може бути закріплена як допустима.

5. Контроль взаємодії контейнерів із SocketCAN.

Відповідно до документації ядра, драйвер CAN-контролера реєструється в мережевій підсистемі Linux як мережевий пристрій, а застосунки взаємодіють із шиною через сокети сімейства AF_CAN; документація описує протоколи сирих сокетів (CAN_RAW) і менеджера трансляції (CAN_BCM), який, зокрема, виконує циклічне передавання кадрів у просторі ядра [25]. Ядро містить також реалізацію

протоколів вищого рівня, зокрема SAE J1939 [26]. Із цього випливають два наслідки для захисту. По-перше, обмеження доступу до файлів пристроїв не охоплює взаємодії з CAN-шиною. По-друге, одна операція передавання в просторі користувача не завжди відповідає одному кадру на шині: після налаштування циклічного завдання BCM ядро продовжує передавання без подальших системних викликів, а транспортні протоколи вищого рівня можуть розбивати повідомлення на кілька кадрів.

З огляду на це обґрунтовано дворівневий контроль. Перший рівень визначає право сервісу на CAN-взаємодію: LSM-хук `socket_create` отримує сімейство, тип і протокол сокета, що дає змогу дозволити конкретному контейнеру лише потрібні протоколи (наприклад, `CAN_RAW` для діагностичного сервісу і жодного для мультимедійного), а хуки `socket_bind` і `socket_sendmsg` — обмежувати перелік доступних CAN-інтерфейсів з урахуванням можливості вказати інтерфейс безпосередньо під час передавання. Другий рівень оцінює відповідність змісту та інтенсивності передавання призначенню сервісу: допустимі ідентифікатори повідомлень, частоту передавання, параметри циклічних завдань BCM. Реалізація другого рівня вимагає перевірки того, які дані доступні в обраній точці інструментування: не можна заздалегідь припускати, що будь-який сокетний хук забезпечує повний аналіз CAN-кадру або охоплює всі шляхи його передавання. Тому контроль змісту доцільно поєднувати зі спостереженням за фактичним трафіком інтерфейсу.

6. Кореляція активності контейнерів із подіями автомобільної мережі.

Системи виявлення вторгнень у CAN-мережах використовують, зокрема, часові характеристики періодичних повідомлень: у CIDS зсув тактових генераторів, оцінений за інтервалами надходження повідомлень, застосовано як ознаку для ідентифікації електронних блоків керування [23]. Такі методи підтверджують інформативність часової поведінки CAN-трафіку, але не встановлюють її зв'язку з процесами Linux-вузла. eBPF дає змогу доповнити мережеві ознаки локальним контекстом: наприклад, зміна частоти повідомлень одночасно з появою неочікуваного процесу в діагностичному контейнері утворює значно змістовніший контекст для аналізу, ніж кожна з цих подій окремо.

Водночас кореляція повинна враховувати межі спостережуваності. Для повідомлення, отриманого від зовнішнього електронного блока керування, локального процесу-відправника не існує; за використання посередницького сервісу або маршрутизації кадрів у ядрі безпосереднім відправником є посередник, а не контейнер, що ініціював запит; для циклічних завдань BCM кадри формує ядро. Тому результат кореляції доцільно подавати з явним розрізненням трьох категорій джерела: встановленого (подію безпосередньо пов'язано з процесом і контейнером), імовірного (зв'язок виведено з часової та змістової близькості подій) і невідомого. Таке розрізнення запобігає хибній атрибуції та дає змогу по-різному використовувати результати: встановлене джерело може бути підставою для автоматичного обмеження, імовірне — для посиленого спостереження, невідоме — лише для реєстрації.

7. Узагальнена архітектура та порядок реагування.

Узагальнення розглянутих механізмів дає змогу запропонувати архітектуру з трьох взаємопов'язаних компонентів (рис. 1). На рівні ядра розміщуються eBPF-програми спостереження і контролю, які виконують короткі перевірки за картами політик і формують мінімально необхідні записи подій. Локальний аналітичний агент у просторі користувача отримує події з кільцевого буфера, збагачує їх метаданими контейнерів, виконує кореляцію з подіями CAN-інтерфейсів і накопичує діагностичну інформацію. Компонент керування політиками: перевірка, версіонування, режими застосування правил, а також керує режимами роботи платформи. Такий поділ дає змогу окремо оцінювати витрати швидкого шляху виконання в ядрі та складнішого аналізу в просторі користувача.

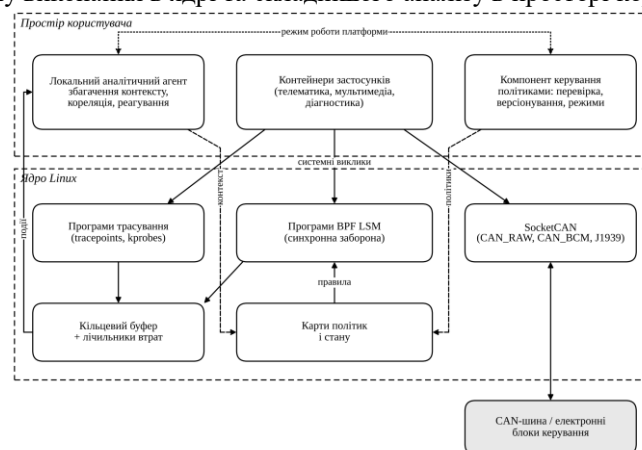


Рис. 1. Узагальнена архітектура захисту автомобільної контейнеризованої платформи на основі eBPF



Реагування доцільно визначати через наслідки конкретного обмеження. Пропонується ступінчаста послідовність дій: реєстрація відхилення, уточнення контексту, заборона конкретної небажаної операції та, за передбачених умов, ізоляція сервісу. Для допоміжного інформаційного застосування припинення роботи може бути прийнятним, тоді як для сервісу, від якого залежать інші функції, рішення про завершення контейнера має враховувати залежності та можливість переходу до резервного або обмеженого режиму.

Поведінку захисту в разі власної відмови також слід визначити заздалегідь. Недоступність аналітичного агента не обов'язково скасовує дію вже завантажених правил ядра, але позбавляє систему кореляції та оновлення контексту. Тому необхідно встановити строк актуальності динамічних дозволів, порядок роботи з невідомими контейнерами та умови відновлення агента; єдина глобальна політика «дозволити все» або «заборонити все» на випадок відмови є недостатньо точною для середовища зі змішаною критичністю функцій.

Окремою вимогою є захист самого механізму eBPF. Дослідження SafeBPF [24] обґрунтовує доцільність додаткової ізоляції eBPF-програм від решти ядра програмними засобами та з використанням апаратного розширення ARM MTE, однак доступність такого механізму на конкретній автомобільній платформі не гарантована. Тому в запропонованій архітектурі необхідно обмежити коло компонентів, які можуть завантажувати програми, змінювати карти політик або від'єднувати механізми контролю, зокрема шляхом контролю системного виклику bpf через відповідний LSM-хук і мінімізації надання можливостей CAP_BPF та CAP_SYS_ADMIN.

8. Критерії та порядок оцінювання.

Оцінювання засобів захисту для автомобільної платформи має враховувати не лише середнє навантаження, а й поведінку за несприятливого поєднання подій, а якість виявлення — визначатися через наслідки для функцій транспортного засобу. Запропоновані критерії згруповано в таблиці 5.

Таблиця 5

Критерії оцінювання eBPF-засобів захисту автомобільних контейнеризованих середовищ

Група	Показники	Призначення
Продуктивність	Використання процесора і пам'яті; затримка контрольованих операцій; вплив на строки виконання прикладних завдань	Оцінювання витрат швидкого шляху та впливу на часові вимоги
Повнота спостереження	Частка зареєстрованих подій; частка втрачених записів; частка подій із невизначеним контекстом	Оцінювання достовірності телеметрії
Якість виявлення	Повнота, точність, час до формування рішення; кількість хибних тривог за годину роботи в кожному режимі	Оцінювання придатності до експлуатації
Наслідки реагування	Кількість необґрунтованих блокувань легітимних операцій; вплив на залежні сервіси	Оцінювання безпечності автоматичного реагування
Стійкість	Поведінка за переповнення буфера, втрати метаданих і відмови агента	Оцінювання поведінки в несприятливих умовах

Експериментально спостережуване максимальне значення затримки слід відрізняти від доведеної верхньої межі: серія вимірювань не встановлює гарантії для всіх можливих станів системи. Так само висока точність класифікації на окремому наборі даних не є достатньою підставою для висновку про прийнятність автоматичного реагування. Перевірку доцільно проводити поетапно: спочатку оцінити повноту збирання подій і правильність ідентифікації контейнерів; далі — правила в режимі спостереження, включно зі сценаріями оновлення, перезапуску сервісів і технічного обслуговування; лише після цього — блокування вибраних операцій та його вплив на залежні компоненти. Окремими сценаріями випробувань мають бути одночасний запуск контейнерів, інтенсивний мережевий і CAN-обмін, переповнення буфера та відмова аналітичного агента.

Отримані результати дають підстави розглядати eBPF як основу для поєднання контекстного моніторингу та вибіркового контролю операцій автомобільних Linux-контейнерів. Межі практичного використання такого підходу визначатимуться тим, за яких конфігурацій платформи він забезпечує прийнятну якість виявлення без порушення часових і функціональних вимог, що потребує подальшої експериментальної перевірки.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Проведене дослідження показало, що технологія eBPF придатна як основа захисту автомобільних контейнеризованих середовищ під час виконання: вона забезпечує низькорівневий моніторинг процесів, системних викликів, файлових операцій і мережевої активності контейнерів без модифікації вихідного



коду ядра та завантаження модулів ядра, а механізм BPF LSM — синхронну заборону окремих операцій. Основні результати роботи такі: 1) систематизовано функції eBPF (моніторинг, виявлення, контроль доступу, контроль SocketCAN, кореляція, реагування) та визначено головне обмеження кожної з них в автомобільному середовищі; 2) обґрунтовано дворівневий контроль взаємодії контейнерів із SocketCAN — окремо права використання CAN-інтерфейсу та відповідності змісту й інтенсивності передавання призначенню сервісу, з урахуванням циклічного передавання засобами CAN Broadcast Manager; 3) запропоновано формувати профілі поведінки для окремих режимів роботи платформи з керуванням переходами між ними через довірений компонент; 4) запропоновано під час кореляції активності контейнера з CAN-трафіком явно розрізняти встановлене, імовірне та невідоме джерело подій; 5) запропоновано архітектуру з трьох компонентів і поетапну методику оцінювання з показниками, що відображають наслідки для автомобіля: хибні тривоги за годину роботи, необґрунтовані блокування, втрати телеметрії, вплив на часові характеристики прикладних завдань. Водночас запропоновані рішення мають концептуальний характер і потребують експериментальної перевірки. Їх застосування обмежене обчислювальними ресурсами платформи, вимогами до передбачуваної затримки, сумісністю версій ядра, вимогами функціональної безпеки та ризиком негативного впливу автоматичного реагування на критичні функції транспортного засобу. Подальші дослідження доцільно спрямувати на реалізацію прототипу на апаратній платформі автомобільного класу, вимірювання накладних витрат і втрат телеметрії під навантаженням, оцінювання якості кореляції процесів із CAN-повідомленнями та розроблення контрольованих механізмів локального реагування на виявлені загрози.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Security Symposium (USENIX Security 11)*. USENIX Association. <https://www.usenix.org/conference/usenix-security-11/comprehensive-experimental-analyses-automotive-attack-surfaces>
2. Gwak, S., Doan, T.-P., & Jung, S. (2023). Container instrumentation and enforcement system for runtime security of Kubernetes platform with eBPF. *Intelligent Automation & Soft Computing*, 37(2), 1773–1786. <https://doi.org/10.32604/iasc.2023.039565>
3. Findlay, W., Barrera, D., & Somayaji, A. (2021). *BPFContain: Fixing the soft underbelly of container security* (arXiv:2102.06972). arXiv. <https://doi.org/10.48550/arXiv.2102.06972>
4. Jarkas, O., Ko, R., Dong, N., & Mahmud, R. (2025). A container security survey: Exploits, attacks, and defenses. *ACM Computing Surveys*, 57(7). <https://doi.org/10.1145/3715001>
5. Jolak, R., Rosenstatter, T., Mohamad, M., Strandberg, K., Sangchoolie, B., Nowdehi, N., & Scandariato, R. (2022). CONSERVE: A framework for the selection of techniques for monitoring containers security. *Journal of Systems and Software*, 186, Article 111158. <https://doi.org/10.1016/j.jss.2021.111158>
6. Kifor, C. V., & Popescu, A. (2024). Automotive cybersecurity: A survey on frameworks, standards, and testing and monitoring technologies. *Sensors*, 24(18), Article 6139. <https://doi.org/10.3390/s24186139>
7. Kim, R., Ryu, J., Kim, S., Lee, S., & Kim, S. (2025). Detecting cryptojacking containers using eBPF-based security runtime and machine learning. *Electronics*, 14(6), Article 1208. <https://doi.org/10.3390/electronics14061208>
8. Lampe, B., & Meng, W. (2023). Intrusion detection in the automotive domain: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 25(4), 2356–2426. <https://doi.org/10.1109/COMST.2023.3309864>
9. Lampe, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications. *Expert Systems with Applications*, 221, Article 119771. <https://doi.org/10.1016/j.eswa.2023.119771>
10. Lim, S. Y., Stelea, B., Han, X., & Pasquier, T. (2021). *Secure namespaces kernel audit for containers* (arXiv:2111.02481). arXiv. <https://doi.org/10.48550/arXiv.2111.02481>
11. Lin, X., Lei, L., Wang, Y., Jing, J., Sun, K., & Zhou, Q. (2018). A measurement study on Linux container security: Attacks and countermeasures. In *Proceedings of the 34th Annual Computer Security Applications Conference* (pp. 418–429). ACM. <https://doi.org/10.1145/3274694.3274720>
12. Lokman, S.-F., Othman, A. T., & Abu-Bakar, M.-H. (2019). Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP Journal on Wireless Communications and Networking*, 2019, Article 184. <https://doi.org/10.1186/s13638-019-1484-3>
13. Nam, J., Lee, S., Porras, P., Yegneswaran, V., & Shin, S. (2023). Secure inter-container communications using XDP/eBPF. *IEEE/ACM Transactions on Networking*, 31(2), 934–947.



- <https://doi.org/10.1109/TNET.2022.3206781>
14. Pan, F., Rickert, M., Betz, T., Wen, L., Lin, J., Petrovic, N., Lienkamp, M., & Knoll, A. (2024). Toward software-defined vehicles: From model-based engineering to virtualization-based deployment. *IEEE Access*, 12, 192127–192145. <https://doi.org/10.1109/ACCESS.2024.3512002>
 15. Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., & Cheah, M. (2023). AI-based intrusion detection systems for in-vehicle networks: A survey. *ACM Computing Surveys*, 55(11), Article 237. <https://doi.org/10.1145/3570954>
 16. Shin, H., Jo, M., Yoo, H., Lee, Y., & Tak, B. (2024). A technique for accurate detection of container attacks with eBPF and AdaBoost. *Journal of the Korea Society of Computer and Information*, 29(6), 39–51. <https://doi.org/10.9708/jksci.2024.29.06.039>
 17. Young, C., Zambreno, J., Olufowobi, H., & Bloom, G. (2019). Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6), 48–55. <https://doi.org/10.1109/MDAT.2019.2899062>
 18. International Organization for Standardization. (2021). *Road vehicles — Cybersecurity engineering* (ISO/SAE Standard No. 21434:2021). <https://www.iso.org/standard/70918.html>
 19. International Organization for Standardization. (2018). *Road vehicles — Functional safety — Part 1: Vocabulary* (ISO Standard No. 26262-1:2018). <https://www.iso.org/standard/68383.html>
 20. The kernel development community. (n.d.). *eBPF verifier*. The Linux Kernel documentation. Retrieved October 14, 2026, from <https://docs.kernel.org/bpf/verifier.html>
 21. The kernel development community. (n.d.). *LSM BPF programs*. The Linux Kernel documentation. Retrieved October 14, 2026, from https://docs.kernel.org/bpf/prog_lsm.html
 22. The kernel development community. (n.d.). *BPF ring buffer*. The Linux Kernel documentation. Retrieved October 14, 2026, from <https://docs.kernel.org/bpf/ringbuf.html>
 23. Cho, K.-T., & Shin, K. G. (2016). Fingerprinting electronic control units for vehicle intrusion detection. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)* (pp. 911–927). USENIX Association. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho>
 24. Lim, S. Y., Prasad, T., Han, X., & Pasquier, T. (2024). *SafeBPF: Hardware-assisted defense-in-depth for eBPF kernel extensions* (arXiv:2409.07508). arXiv. <https://doi.org/10.48550/arXiv.2409.07508>
 25. The kernel development community. (n.d.). *SocketCAN — Controller Area Network*. The Linux Kernel documentation. Retrieved October 14, 2026, from <https://docs.kernel.org/networking/can.html>
 26. The kernel development community. (n.d.). *J1939 documentation*. The Linux Kernel documentation. Retrieved October 14, 2026, from <https://docs.kernel.org/networking/j1939.html>

**Anastasiia Zhuravchak**

PhD, Teaching Assistant of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0002-8196-7963
anastasiia.y.tolkachova@lpnu.ua

Danyil Zhuravchak

PhD, Associate Professor of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0003-4989-0203
danyil.y.zhuravchak@lpnu.ua

Yurii Zhuravchak

PhD student of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0009-0003-2378-5365
yurii.zhuravchak.asp.2026@lpnu.ua

USING eBPF TO PROTECT AUTOMOTIVE CONTAINERIZED ENVIRONMENTS

Abstract. Software-defined vehicles increasingly rely on Linux platforms, centralized compute nodes and containerized applications. A shared operating system kernel creates risks of isolation breaches, privilege escalation and unauthorized access of containers to vehicle resources and in-vehicle networks. The aim of this work is to systematize approaches to using eBPF for runtime monitoring, threat detection and security policy enforcement in automotive containerized environments. The methodology is based on an analysis of scientific publications and Linux kernel documentation and on a comparison of protection mechanisms for containers and automotive networks. The functions of eBPF (monitoring, detection, access control, SocketCAN control, correlation, response) are systematized, and the main limitation of each in the automotive setting is identified. The paper substantiates the separation of event logging from synchronous blocking via BPF LSM; two-level control of SocketCAN interaction that separately checks the right to use a CAN interface and the conformity of transmissions with the service's purpose; mode-specific behavior profiles; and an explicit distinction between established, probable and unknown event origin when correlating container activity with CAN traffic. A three-component architecture (kernel-level mechanisms, a local analytics agent, a policy management component) and evaluation criteria are proposed: event collection completeness, telemetry loss rate, false alarms per operating hour, unjustified blocks and impact on application timing. The practical value lies in defining requirements for protection tools and a staged procedure for their validation on automotive Linux platforms. Future work concerns experimental validation of the proposed solutions on automotive-grade hardware.

Keywords: eBPF; cybersecurity; containerization; Linux; monitoring; intrusion detection; SocketCAN; CAN bus; BPF LSM; software-defined vehicle.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Security Symposium (USENIX Security 11)*. USENIX Association.
2. Gwak, S., Doan, T.-P., & Jung, S. (2023). Container instrumentation and enforcement system for runtime security of Kubernetes platform with eBPF. *Intelligent Automation & Soft Computing*, 37(2), 1773–1786. <https://doi.org/10.32604/iasc.2023.039565>
3. Findlay, W., Barrera, D., & Somayaji, A. (2021). *BPFContain: Fixing the soft underbelly of container security* (arXiv:2102.06972). arXiv. <https://doi.org/10.48550/arXiv.2102.06972>
4. Jarkas, O., Ko, R., Dong, N., & Mahmud, R. (2025). A container security survey: Exploits, attacks, and defenses. *ACM Computing Surveys*, 57(7). <https://doi.org/10.1145/3715001>
5. Jolak, R., Rosenstatter, T., Mohamad, M., Strandberg, K., Sangchoolie, B., Nowdehi, N., & Scandariato, R. (2022). CONSERVE: A framework for the selection of techniques for monitoring containers security.



- Journal of Systems and Software*, 186, Article 111158. <https://doi.org/10.1016/j.jss.2021.111158>
6. Kifor, C. V., & Popescu, A. (2024). Automotive cybersecurity: A survey on frameworks, standards, and testing and monitoring technologies. *Sensors*, 24(18), Article 6139. <https://doi.org/10.3390/s24186139>
 7. Kim, R., Ryu, J., Kim, S., Lee, S., & Kim, S. (2025). Detecting cryptojacking containers using eBPF-based security runtime and machine learning. *Electronics*, 14(6), Article 1208. <https://doi.org/10.3390/electronics14061208>
 8. Lampe, B., & Meng, W. (2023). Intrusion detection in the automotive domain: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 25(4), 2356–2426. <https://doi.org/10.1109/COMST.2023.3309864>
 9. Lampe, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications. *Expert Systems with Applications*, 221, Article 119771. <https://doi.org/10.1016/j.eswa.2023.119771>
 10. Lim, S. Y., Stelea, B., Han, X., & Pasquier, T. (2021). *Secure namespaced kernel audit for containers* (arXiv:2111.02481). arXiv. <https://doi.org/10.48550/arXiv.2111.02481>
 11. Lin, X., Lei, L., Wang, Y., Jing, J., Sun, K., & Zhou, Q. (2018). A measurement study on Linux container security: Attacks and countermeasures. In *Proceedings of the 34th Annual Computer Security Applications Conference* (pp. 418–429). ACM. <https://doi.org/10.1145/3274694.3274720>
 12. Lokman, S.-F., Othman, A. T., & Abu-Bakar, M.-H. (2019). Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP Journal on Wireless Communications and Networking*, 2019, Article 184. <https://doi.org/10.1186/s13638-019-1484-3>
 13. Nam, J., Lee, S., Porras, P., Yegneswaran, V., & Shin, S. (2023). Secure inter-container communications using XDP/eBPF. *IEEE/ACM Transactions on Networking*, 31(2), 934–947. <https://doi.org/10.1109/TNET.2022.3206781>
 14. Pan, F., Rickert, M., Betz, T., Wen, L., Lin, J., Petrovic, N., Lienkamp, M., & Knoll, A. (2024). Toward software-defined vehicles: From model-based engineering to virtualization-based deployment. *IEEE Access*, 12, 192127–192145. <https://doi.org/10.1109/ACCESS.2024.3512002>
 15. Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., & Cheah, M. (2023). AI-based intrusion detection systems for in-vehicle networks: A survey. *ACM Computing Surveys*, 55(11), Article 237. <https://doi.org/10.1145/3570954>
 16. Shin, H., Jo, M., Yoo, H., Lee, Y., & Tak, B. (2024). A technique for accurate detection of container attacks with eBPF and AdaBoost. *Journal of the Korea Society of Computer and Information*, 29(6), 39–51. <https://doi.org/10.9708/jksci.2024.29.06.039>
 17. Young, C., Zambreno, J., Olufowobi, H., & Bloom, G. (2019). Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6), 48–55. <https://doi.org/10.1109/MDAT.2019.2899062>
 18. International Organization for Standardization. (2021). *Road vehicles — Cybersecurity engineering* (ISO/SAE Standard No. 21434:2021). <https://www.iso.org/standard/70918.html>
 19. International Organization for Standardization. (2018). *Road vehicles — Functional safety — Part 1: Vocabulary* (ISO Standard No. 26262-1:2018). <https://www.iso.org/standard/68383.html>
 20. The kernel development community. (n.d.). *eBPF verifier*. The Linux Kernel documentation.
 21. The kernel development community. (n.d.). *LSM BPF programs*. The Linux Kernel documentation.
 22. The kernel development community. (n.d.). *BPF ring buffer*. The Linux Kernel documentation.
 23. Cho, K.-T., & Shin, K. G. (2016). Fingerprinting electronic control units for vehicle intrusion detection. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)* (pp. 911–927). USENIX Association. <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho>
 24. Lim, S. Y., Prasad, T., Han, X., & Pasquier, T. (2024). *SafeBPF: Hardware-assisted defense-in-depth for eBPF kernel extensions* (arXiv:2409.07508). arXiv. <https://doi.org/10.48550/arXiv.2409.07508>
 25. The kernel development community. (n.d.). *SocketCAN — Controller Area Network*. The Linux Kernel documentation. Retrieved October 14, 2026, from <https://docs.kernel.org/networking/can.html>
 26. The kernel development community. (n.d.). *J1939 documentation*. The Linux Kernel documentation.

Отримано редакцією журналу / Received: 20.08.26

Прорецензовано / Revised: 30.08.26

Схвалено до друку / Accepted: 26.10.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.



КІБЕРБЕЗПЕКА: освіта, наука, техніка

№ 3 (35), 2026

CYBERSECURITY:
EDUCATION, SCIENCE, TECHNIQUE

Спеціальний випуск ISSN 2663 – 4023
Special issue