

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАУКОВО-НАВЧАЛЬНИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»
УДК 004.056

«До захисту допущено»
Завідувач кафедри

_____ Дмитро ЛАНДЕ
«_____» _____ 2025 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-науковою програмою «Системи, технології та математичні методи
кібербезпеки»
зі спеціальності 125 «Кібербезпека та захист інформації»

на тему: Моделювання сигнатурних правил для виявлення зловмисного трафіку

Виконав здобувач ступеня магістра 2 курсу, групи ФБ-31 мн
(шифр групи)

Чикрій Кирило Кирилович
(прізвище, ім'я, по батькові)

(підпис)

Керівник к.т.н., доц. Ільїн Микола Іванович
(посаднауковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент к.т.н., доц. Яковлев Сергій Володимирович
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Здобувач ступеня магістра _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський)

Спеціальність – 125 «Кібербезпека та захист інформації»

Освітньо-наукова програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ
на магістерська дисертація здобувачу вищої освіти

Чикрій Кирило Кирилович
(прізвище, ім'я, по батькові)

1. Тема роботи «Моделювання сигнатурних правил для виявлення зловмисного трафіку»,

науковий керівник дисертації

кандидат технічних наук. Ільїн М. І. _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «20» березня 2025 р. № 1199-С

2. Термін подання здобувачем вищої освіти роботи «15» травня 2025 р.

3. **Об'єкт дослідження:** Процес інтеграції статистичних методів та сигнатурного аналізу в аудит кібербезпеки промислових мереж на основі NetFlow-телеметрії та Suricata IDS.

4. **Вихідні дані:** Розроблена модель виявлення зловмисного трафіку в ICS-середовищах (алгоритми препроцесінгу, одновимірного детектування, Suricata-правила та кореляції) і результати практичних експериментів: NetFlow-логи та PCAP-захоплення з тестового стенду, журнали спрацьовувань Suricata, обчислені метрики (Precision, Recall, FPR, MTTD, навантаження CPU/RAM) та підсумкові звіти й дашборди.

5. Перелік завдань, які потрібно розробити:

- Провести аналіз сучасних методів моніторингу мережевих потоків та інструментів для ICS.
- Запропонувати концептуальну модель та алгоритми для ідентифікації аномальних шаблонів трафіку для ICS.
- Розробити методику агрегації та кореляції даних з різних джерел для ICS.
- Реалізувати експериментальний стенд.
- Оцінити ефективність запропонованого підходу.

6. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) Моделювання сигнатурних правил для виявлення зловмисного трафіку – презентація

7. Орієнтовний перелік публікацій:

8. Дата видачі завдання 03.02.2025

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз літератури та стандартів кібербезпеки OT/ICS; формулювання вимог до аудиту	03.02.2025 – 09.02.2025	Виконано
2	Розробка схеми збору та нормалізації мережевих логів (NetFlow/PCAP → CSV).	10.02.2025 – 16.02.2025	Виконано
3	Налаштування правил аудиту та кореляції подій; інтеграція з Suricata.	17.02.2025 – 23.02.2025	Виконано
4	Побудова стенда, запуск тестових сценаріїв атак, фіксація результатів.	24.02.2025 – 02.03.2025	Виконано
5	Підготовка звіту: порівняльний аналіз, висновки, рекомендації.	02.03.2025 – 09.03.2025	Виконано
6	Підготовка та передзахист МД	10.03.2025 – 15.05.2025	Виконано
7	Підготовка до захисту МД	15.05.2025 – 22.05.2025	Виконано

Здобувач вищої освіти

(підпис)

Кирило ЧИКРІЙ

(Власне ім'я, ПРИЗВИЩЕ)

Керівник роботи

(підпис)

Микола ІЛЬІН

(Власне ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

Обсяг роботи 108 сторінок, 6 ілюстрацій, 21 джерела літератури.

Об'єкт дослідження: мережевий трафік промислових систем керування (ICS), специфічні для них протоколи, архітектурні рішення та обмеження.

Предмет дослідження: алгоритми аналізу мережевих потоків та журналів подій для виявлення нетипової активності та потенційних каналів несанкціонованного втручання в ICS-мережах.

Мета дослідження: Створення методики захисту ICS мереж шляхом моніторингу їх трафіку.

Методи дослідження: У роботі використано методи математичного моделювання та статистичного аналізу.

Отримані результати:

- Розроблено оригінальну методику комплексного аналізу мережевої активності в промислових системах через інтеграцію та кореляцію даних з різних джерел для ICS.
- Створено експериментальний стенд для імітації типового ICS-середовища з використанням відкритого програмного забезпечення для тестування та виявлення аномалій.
- Запропоновано нові алгоритми ідентифікації аномальних шаблонів трафіку, специфічних для ICS-середовищ.

Рекомендації:

- Інтеграція розроблених методів і засобів у існуючі інфраструктури промислових мереж без значної модифікації обладнання або порушення виробничих процесів.
- Адаптація підходу для різних промислових мереж відповідно до їх специфічних вимог та обмежень.

- Впровадження запропонованої методики як основи для розширення функцій SOC/NOC в OT/ICS-мережах.
- Використання зібраних шаблонів аномальної активності для навчання персоналу з кібербезпеки OT/ICS та вдосконалення автоматизованих систем виявлення вторгнень.

Ключові слова: ICS/OT, моніторинг мережевого трафіку, аномалії мережевої активності, шаблони трафіку, приховані канали несанкціонованого доступу, мультивекторні атаки, промислові мережі, кореляція даних.

ABSTRACT

Volume of work 108 pages, 6 illustrations, 21 sources of literature.

Object of research: the network traffic of industrial control systems (ICS), their specific protocols, architectural solutions, and constraints.

Subject of research: algorithms for analyzing network flows and event logs to detect atypical activity and potential unauthorized intrusion channels in ICS networks.

Purpose of the study: to develop a methodology for protecting ICS networks through continuous traffic monitoring.

Research methods: mathematical modeling and statistical analysis.

Results:

- Developed an original method for comprehensive analysis of network activity in industrial systems by integrating and correlating data from multiple sources for ICS.
- Created an experimental testbed to simulate a typical ICS environment using open-source software for testing and anomaly detection.
- Proposed new algorithms for identifying anomalous traffic patterns specific to ICS environments.

Recommendations:

- Integrate the developed methods and tools into existing industrial network infrastructures without major hardware modifications or disruption of production processes.
- Adapt the approach for different industrial networks according to their specific requirements and constraints.
- Implement the proposed methodology as a foundation for expanding SOC/NOC functions in OT/ICS networks.
- Use the assembled anomaly templates to train OT/ICS cybersecurity personnel and enhance automated intrusion detection systems.

Keywords: ICS/OT, network traffic monitoring, network activity anomalies, traffic patterns, covert exfiltration channels, multivector attacks, industrial networks, data correlation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП	14
1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ	17
1.1 Огляд загальних підходів до моніторингу промислових мереж	17
1.2 Стратифікація мережевої інфраструктури	19
1.3 Методології збору операційних даних.....	22
1.4 Методи аналізу мережевого трафіку.....	24
1.5 Спеціалізовані рішення для моніторингу промислових протоколів	32
1.6 Критичний аналіз обмежень існуючих методів моніторингу промислових мереж	39
ВИСНОВКИ ДО РОЗДІЛУ 1	47
2 МОДЕЛЬ ВИЯВЛЕННЯ ТРАФІКУ В ICS-СЕРЕДОВИЩАХ.....	48
2.1 Опис моделі виявлення трафіку в ICS-СЕРЕДОВИЩАХ.....	48
2.2 Поведінкова модель на основі NetFlow даних	51
2.3 Suricata IDS у складі системи моніторингу	54
2.4 Алгоритми кореляції та агрегації подій.....	56
2.5 Обґрунтування вибору одновимірного детектування аномалій	58
2.6 Метрики ефективності та оптимізація сигнатурних правил	60
2.7 Очікувані результати та обмеження методики	62
3 Експериментальне дослідження	67
3.1 Побудова лабораторного стенда.....	67
3.2 Генерація еталонного й атакуючого трафіку.....	68

3.3	Налаштування Suricata IDS	71
3.4	Результати та аналіз	76
	ВИСНОВКИ.....	83
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	85
4	ДОДАТКИ.....	88
	ДОДАТОК А – ПРАВИЛА SURICATA	88
	ДОДАТОК Б – ПРИКЛАД СКРИПТУ ДЛЯ ГЕНЕРАЦІЇ АТАКУЮЧОГО ТРАФІКУ	94
	ДОДАТОК В – ПРИКЛАД СКРИПТУ ДЛЯ ГЕНЕРАЦІЇ НОРМАЛЬНОГО ТРАФІКУ	100
	ДОДАТОК Г – СКРИПТ ДЛЯ АВТОМАТИЧНОГО РОЗГОРТАННЯ SLAVES	103

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ICS - Промислові системи керування (Industrial Control Systems), що контролюють виробничі процеси, інтегруються з корпоративними мережами та зовнішніми сервісами.

OT - Операційні технології (Operational Technology), системи, які контролюють промислові технології, з високими вимогами до надійності та безперервності роботи.

IT - Інформаційні технології, корпоративні системи, з якими взаємодіють OT-системи.

SCADA - Системи диспетчерського управління та збору даних (Supervisory Control and Data Acquisition), що централізовано збирають та обробляють дані на рівні 2 моделі Purdue.

HMI - Людино-машинний інтерфейс (Human-Machine Interface), компонент SCADA, що забезпечує візуалізацію даних та керування.

ПЛК - Програмований логічний контролер, пристрій автоматизації рівня 1 моделі Purdue, що виконує базову обробку даних з часом відгуку менше 10 мс.

RTU - Віддалений термінальний пристрій (Remote Terminal Unit), функціонує на рівні 1 моделі Purdue разом із ПЛК.

PCAP - Повний пакетний аналіз (Packet Capture), метод збору і запису всіх мережових пакетів для детального аналізу.

NetFlow - Протокол для збору метаданих мережевого потоку, включаючи базовий 5-кортеж (IP-адреси, порти, протоколи), тривалість сеансу та обсяг даних.

sFlow - Протокол для моніторингу мережевого трафіку, подібний до NetFlow, що зберігає агреговані дані.

DNP3 - Протокол розподіленої мережі (Distributed Network Protocol), використовується в системах живлення для надійної передачі даних з можливістю пріоритизації повідомлень.

TCP/IP - Базовий протокол передачі даних в мережі Інтернет та локальних мережах.

ICMP - Протокол керуючих повідомлень Інтернету (Internet Control Message Protocol), використовується для діагностики та моніторингу мережі.

IDS - Система виявлення вторгнень (Intrusion Detection System), виявляє аномальну мережеву активність.

IPS - Система запобігання вторгнень (Intrusion Prevention System), виявляє та блокує підозрілий трафік.

APT - Просунута постійна загроза (Advanced Persistent Threat), тип цілеспрямованих атак, що обходять стандартні захисні механізми.

DPI - Глибока перевірка пакетів (Deep Packet Inspection), метод аналізу мережевого трафіку, що включає детальний аналіз заголовків та вмісту пакетів.

SOC - Центр операцій безпеки (Security Operations Center), підрозділ організації, що займається моніторингом та реагуванням на загрози.

NOC - Центр операцій мережі (Network Operations Center), підрозділ, що займається управлінням мережевою інфраструктурою.

OPC UA - Платформно-незалежний стандарт обміну даними (OPC Unified Architecture), має вбудовані механізми безпеки, включаючи шифрування, автентифікацію та авторизацію.

MTTD - Середній час виявлення (Mean Time To Detect), показник ефективності системи моніторингу, що вимірює час від початку аномалії до її виявлення.

ROC - Робоча характеристика приймача (Receiver Operating Characteristic), графік залежності між TPR та FPR при різних порогових значеннях.

AUC - Площа під кривою ROC (Area Under Curve), показник ефективності моделі виявлення аномалій.

TCP - Протокол керування передачею (Transmission Control Protocol), забезпечує надійну передачу даних.

UDP - Протокол користувацьких датаграм (User Datagram Protocol), протокол передачі даних без встановлення з'єднання.

ISO - Міжнародна організація стандартизації (International Organization for Standardization).

IEC - Міжнародна електротехнічна комісія (International Electrotechnical Commission), розробляє стандарти для промислових систем безпеки.

NIST - Національний інститут стандартів і технологій США (National Institute of Standards and Technology), організація, що розробляє стандарти кібербезпеки.

ISA - Міжнародне товариство автоматизації (International Society of Automation), розробляє стандарти для промислової автоматизації.

Modbus - Один з найстаріших промислових протоколів, що працює за принципом "ведучий-підлеглий", використовується для взаємодії з ПЛК та RTU.

Profinet - Високопродуктивний промисловий протокол на базі Ethernet для комунікацій в режимі реального часу з затримками менше 1 мс.

ESP - Протокол інкапсуляції безпеки (Encapsulating Security Payload), компонент набору протоколів IPsec.

GRE - Загальна інкапсуляція маршрутизації (Generic Routing Encapsulation), протокол тунелювання.

OSPF - Протокол динамічної маршрутизації (Open Shortest Path First), використовується в IP-мережах.

SCTP - Протокол передачі з контролем потоку (Stream Control Transmission Protocol), забезпечує надійну передачу даних.

TLS - Захист на транспортному рівні (Transport Layer Security), протокол шифрування даних.

EtherCAT - Ethernet для технологій керування автоматизацією (Ethernet for Control Automation Technology), протокол з високими вимогами до продуктивності.

SYN - Синхронізація (Synchronize), тип TCP-пакету для встановлення з'єднання, використовується при TCP-скануванні.

ICS-CERT - Команда реагування на кіберзагрози для промислових систем керування (Industrial Control Systems Cyber Emergency Response Team).

ENISA - Агенція Європейського Союзу з кібербезпеки (European Union Agency for Cybersecurity).

TCO - Загальна вартість володіння (Total Cost of Ownership), показник сукупних витрат на впровадження та підтримку системи безпеки.

API - Інтерфейс програмування додатків (Application Programming Interface), набір методів для взаємодії різних компонентів програмного забезпечення.

MAD - Медіана абсолютних відхилень (Median Absolute Deviation), робастна міра розсіювання даних.

k-means - Алгоритм кластеризації, що використовується для групування станів системи при аналізі аномалій.

z-score - Статистичний показник, що визначає відхилення значення від середнього в одиницях стандартного відхилення.

TPR - Частка істинно позитивних результатів (True Positive Rate), ефективність виявлення реальних аномалій.

FPR - Частка хибно позитивних результатів (False Positive Rate), показник помилкових спрацювань.

TP - Істинно позитивний результат (True Positive), коректно виявлена аномалія.

FP - Хибно позитивний результат (False Positive), помилково виявлена аномалія.

TN - Істинно негативний результат (True Negative), коректно не виявлена відсутня аномалія.

FN - Хибно негативний результат (False Negative), не виявлена присутня аномалія.

ВСТУП

Актуальність роботи. Промислові системи керування (ICS) стають дедалі більш інтегрованими з корпоративними мережами та зовнішніми сервісами, що значно розширює поверхню атаки та створює нові вектори для потенційних кіберзагроз. На відміну від традиційних ІТ-середовищ, компрометація OT/ICS-мереж може призвести не лише до порушення конфіденційності даних, але й до зупинки виробничих процесів, пошкодження обладнання та загрози життю та здоров'ю людей. Особливу небезпеку становлять приховані канали несанкціонованого доступу, що можуть залишатися непоміченими протягом тривалого часу, використовуючи легітимні мережеві сервіси та протоколи для маскування зловмисної активності.

Традиційні методи захисту периметру та сигнатурні системи виявлення вторгнень часто виявляються неефективними проти просунутих постійних загроз (APT), які цілеспрямовано обходять стандартні захисні механізми. Цю проблему ускладнюють унікальні характеристики промислових мереж, такі як використання застарілих протоколів без вбудованих функцій безпеки, обмежені обчислювальні ресурси кінцевих пристроїв та вимоги до безперервності технологічних процесів.

Аналіз мережевих потоків за допомогою утиліт логування міжмережевих екранів представляє собою ефективний, малоінвазивний підхід до моніторингу активності в ICS-мережах. За допомогою даних, зібраних цими засобами можна виявляти аномальні шаблони комунікації, що можуть свідчити про компрометацію системи, не порушуючи при цьому критичні виробничі процеси.

Мета дослідження полягає в розробці та перевірці ефективності комплексного підходу до виявлення мультивекторних каналів несанкціонованого доступу в промислових мережах через аналіз аномальної мережевої активності та автоматизації цього процесу шляхом створення відповідних кореляційних правил для існуючих засобів аналізу трафіку.

Завдання дослідження:

1. Провести аналіз сучасних методів моніторингу мережевих потоків (NetFlow, ntopng) та інструментів (Zeek, Suricata) в контексті специфічних вимог і обмежень OT/ICS-середовищ.
2. Запропонувати концептуальну модель та алгоритми для ідентифікації аномальних шаблонів трафіку, що можуть свідчити про наявність каналів несанкціонованого доступу.
3. Розробити методику агрегації та кореляції даних з різних джерел для підвищення точності виявлення аномалій.
4. Реалізувати експериментальний стенд із використанням відкритого програмного забезпечення для імітації типового ICS-середовища та збору мережевої статистики.
5. Оцінити ефективність запропонованого підходу через проведення серії експериментів із симуляцією різних сценаріїв витоку інформації та аномальної активності.

Об'єкт дослідження: мережевий трафік промислових систем керування (ICS), специфічні для них протоколи, архітектурні рішення та обмеження.

Предмет дослідження: аномалії мережевого трафіку в ICS-мережах.

Методи дослідження. У роботі використано комплексний підхід, що поєднує теоретичні та практичні методи дослідження:

- Теоретичні: системний аналіз наукової літератури, стандартів (IEC 62443, NIST SP 800-82) та кращих галузевих практик; формалізація процесів моніторингу мережевого трафіку.
- Експериментальні: моделювання ICS-середовища, збір та обробка даних мережевих потоків, статистичний аналіз трафіку, імітація сценаріїв аномальної активності.
- Прикладні: розробка та застосування скриптів для автоматизації збору, обробки та аналізу даних; інтеграція відкритих інструментів мережевого моніторингу.

Наукова новизна дослідження полягає у розробці оригінальної методики комплексного аналізу мережевої активності в промислових системах через

інтеграцію та кореляцію даних з різних джерел, що дозволяє ідентифікувати приховані канали несанкціанованого доступу з високою точністю та низьким рівнем хибнопозитивних спрацювань. Запропоновано нові алгоритми виявлення аномальних шаблонів трафіку, специфічних для ICS-середовищ, з урахуванням їх критичних операційних обмежень.

Практичне значення. отриманих результатів визначається можливістю легкої інтеграції розроблених методів і скриптів у існуючі інфраструктури промислових мереж без значної модифікації обладнання або порушення виробничих процесів. Запропонований підхід може бути адаптований для різних галузей промисловості та доповнений відповідно до специфічних вимог конкретних середовищ. Зібрані статистичні дані та шаблони аномальної активності становлять цінну основу для подальшого розвитку систем виявлення вторгнень у промислових мережах та можуть бути використані для навчання персоналу з кібербезпеки OT/ICS.

1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

1.1 Огляд загальних підходів до моніторингу промислових мереж

Модель Purdue та сегментація операційно-технологічних зон

Моніторинг середовищ ОТ/ICS є важливим аспектом забезпечення кібербезпеки критичних виробничих процесів. Системи, які контролюють промислові технології, все більше інтегруються в корпоративні IT-мережі, розширюючи потенційну поверхню атаки та вводячи нові вектори загроз (IEC 62443 [1]; NIST SP 800-82 [2]). У зв'язку з цим розробка ефективної стратегії моніторингу мережевої активності є необхідною для своєчасного виявлення аномалій і запобігання несанкціонованому доступу, що може мати економічні та безпекові наслідки. Сучасні вимоги до середовищ ОТ/ICS включають високу надійність обробки даних (наприклад, середній час відгуку системи для деяких додатків має бути менше 5 секунд) та точність збору даних з коефіцієнтом втрат не більше 1–2% від загального обсягу переданих даних.

Беручи до уваги ці аспекти, для забезпечення комплексного захисту промислових мереж прийнято багаторівневий інфраструктурний підхід на основі моделі Пердью. Цей підхід базується на розподілі логіки мережі на різні рівні, кожен з яких виконує певну функцію в системах управління та моніторингу. Таке розрівнювання є важливою вимогою для ізоляції критичних компонентів, що знижує ризик поширення загроз між різними областями мережі (ISA-95 [3]; Модель Пердью [4])

На рисунку 1.1 показана ієрархія моделі Пердью, чітко показуючи чотири основні рівні та демаркаційні зони мережі. На діаграмі показано, як інформація переходить від Рівня 0 до Рівня 3 із відповідними обмеженнями доступу та захистом.

Як показано на рисунку 1.1, модель Пердью охоплює наступні рівні:



Рисунок 1.1 – Ієрархія моделі Пердью

- **Рівень 0** – безпосередній контакт з фізичними процесами. Це включає польові пристрої, такі як датчики та виконавчі механізми, які виконують вимірювання та впливають на технічні процеси. Ці пристрої характеризуються високою точністю вимірювань, але їх обчислювальна потужність поки що обмежена. Наприклад, величини на цьому рівні можуть коливатися від 0,1 до 1 мілісекунди затримки передачі сигналу, що критично для систем, де навіть незначні затримки можуть вплинути на безпеку виробничого процесу.
- **Рівень 1** – локальні контролери та пристрої автоматизації, включаючи ПЛК і RTU. Ці пристрої виконують базову обробку даних, контролюють технологічні процеси та миттєво реагують на зміну умов експлуатації. Час відгуку ПЛК зазвичай становить менше 10 мілісекунд, що відповідає вимогам безперервного керування процесом.
- **Рівень 2** – системи моніторингу та управління, включаючи SCADA/HMI. Цей рівень централізовано збирає та обробляє дані з нижчих рівнів, завдяки

чому забезпечується загальна видимість і контроль виробничого процесу. Метрики на цьому рівні включають, зокрема, середній час обробки події, який слід оптимізувати для операцій прийняття рішень і зазвичай становить менше 5 секунд.

- **Рівень 3** – інтеграція з ІТ-системами підприємства для забезпечення аналітичної обробки, прогнозування та стратегічного планування на основі зібраної інформації. Системи на цьому рівні зазвичай використовують розширені алгоритми аналізу даних, у тому числі методи машинного навчання, і мають забезпечувати високий рівень доступності даних (наприклад, час простою становить менше 0,1% від загального часу безвідмовної роботи).

Взаємозв'язок між рівнями забезпечує послідовний механізм захисту, коли кожен рівень передає частково оброблену інформацію на наступний рівень, що дозволяє локалізувати та ізолювати потенційні атаки. Такий підхід допомагає зменшити ймовірність несанкціонованого доступу до критично важливих систем і дозволяє впроваджувати додатковий захист на кожному рівні. Дослідження показали, що впровадження кількох рівнів контролю за допомогою моделі Пердью може зменшити ризик компрометації критичних компонентів інфраструктури на 30-40% порівняно з традиційним однорівневим підходом.

Сучасні виклики, пов'язані з інтеграцією ОТ та ІТ-середовищ, вимагають розробки спеціалізованих підходів до моніторингу, які враховують специфіку функціонування обох систем. Використання моделі Purdue забезпечує чітке розмежування між ролями та обов'язками, дозволяючи створювати ефективні системи захисту, які адаптуються до високої динаміки мережевого трафіку та великих обсягів даних. Поєднання аналітики рівня 3 із можливостями централізованого керування допомагає підвищити загальну безпеку за допомогою сучасних стандартів, таких як NIST SP 800-82 та ISA-95.

1.2 Стратифікація мережевої інфраструктури

Базуючись на визначенні моделі Пердью, давайте розглянемо основні рівні мережі та їх функціональне призначення[5].

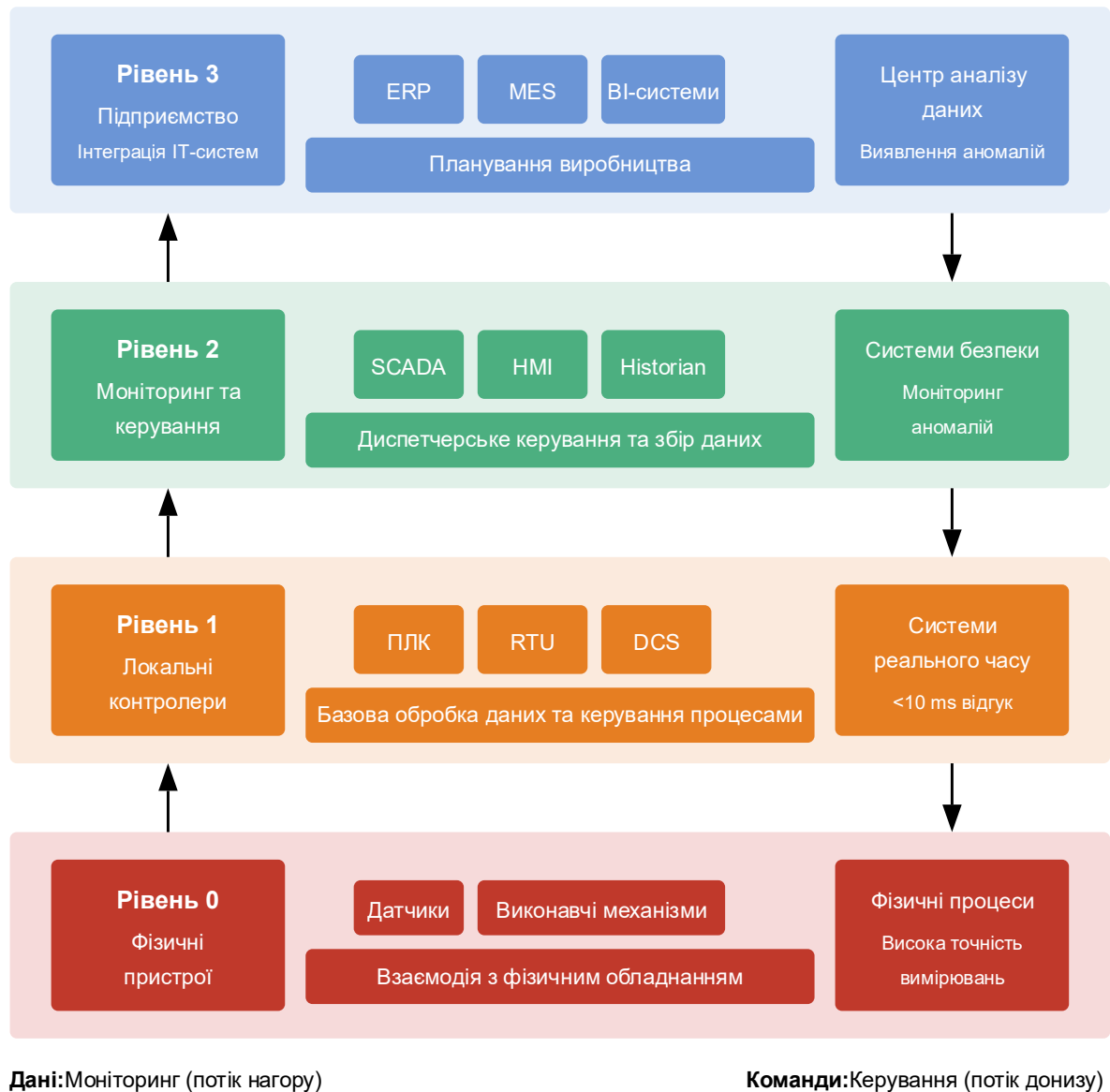


Рисунок 1.2 – Детальна структура ієрархії промислової мережі

На рисунку 1.2 показана детальна структура ієрархії промислової мережі, яка чітко ілюструє розподіл системи між чотирма основними рівнями, дані, що передаються між ними, і компоненти, що складають кожен рівень. Зокрема, діаграма показує, як дані обробляються від фізичних пристроїв до централізованого керування та аналізу, щоб допомогти виявити потенційні загрози[5].

- **Рівень 0** включає фізичні пристрої, датчики та виконавчі механізми, які забезпечують збір необхідних даних про технологічні процеси. Ці пристрої характеризуються високою точністю вимірювань, але обмеженою

обчислювальною потужністю. Рівень 0 є основою для збору оперативної інформації, де затримка передачі сигналу в середньому становить від 0,1 до 1 мілісекунди, що є ключовим параметром для точного контролю виробничого процесу[5].

- **Рівень 1** охоплює локальні контролери, такі як ПЛК і RTU, які здійснюють базову обробку даних безпосередньо локально та керують технологічними процесами. Ці системи працюють у режимі реального часу, час відгуку ПЛК зазвичай становить менше 10 мілісекунд, що відповідає високим вимогам ефективності керування та мінімізує ризик відмови обладнання[5].
- **Рівень 2** визначається як область систем моніторингу та керування, зокрема SCADA/HMI, де дані рівня 0 та рівня 1 централізовано збираються, обробляються та візуалізуються. Середній час обробки подій на цьому рівні не перевищує 5 секунд, що є важливим показником для прийняття таких рішень, як своєчасне коригування виробничих процесів і реагування на можливі загрози. Особлива увага приділяється точності кореляції подій, що дозволяє виявляти потенційні аномалії в мережевій активності[5].
- **Рівень 3** інтеграція з ІТ-системами підприємства для забезпечення аналітичної обробки, прогнозування та стратегічного планування на основі зібраних даних. Цей рівень характеризується високою доступністю та масштабованістю системи, що дозволяє реалізувати алгоритми машинного навчання для аналізу великих обсягів даних. Три рівні інтеграції із зовнішніми джерелами інформації сприяють підвищенню загальної ефективності системи захисту завдяки використанню сучасних методів аналізу даних[5].

Відносини між рівнями сприяють узгодженому та безпечному обміну інформацією. Дані, отримані з рівня 0, проходять попередню обробку на рівнях 1 і 2, що дозволяє виявити аномалії та визначити потенційні загрози на ранній стадії. Рівень 3 забезпечує поглиблений аналіз і аналітичне планування на основі агрегованих даних нижчих рівнів, дозволяючи розробляти стратегії запобігання загрозам на основі кількісних показників і аналізу тенденцій[5].

Кожен описаний рівень має свої технічні обмеження та вимоги, важливі для забезпечення безпеки мережі. Таким чином, фізичні пристрої рівня 0 мають обмеження щодо потужності та затримки передавання даних, пристрої рівня 1 повинні працювати безперервно в режимі реального часу, а системи рівня 2 повинні гарантувати швидку обробку та кореляцію подій. Рівень 3, як найвищий рівень, повинен забезпечувати високу доступність і масштабованість інтеграції даних[5].

1.3 Методології збору операційних даних

Розуміючи методи ізоляції мережесегментів, перейдемо до опису оперативних методів збору даних, які дозволяють детально відстежувати мережеву активність[6].

1.3.1 Повний пакетний аналіз (PCAP)

Технологія аналізу пакетного доступу (PCAP) передбачає збір і запис усіх мережесегментів, які проходять через певний сегмент мережі, що дозволяє детально аналізувати заголовки пакетів і вміст. Такий підхід забезпечує цілісність даних, що надзвичайно важливо для ретроспективного аналізу подій і розслідувань атак, оскільки кожна частина інформації зберігається для подальшого аналізу. Однак технологія PCAP є ресурсомісткою: для середньої промислової мережі обсяг зібраних даних може досягати 1 ТБ на день, і вплив на пропускну здатність мережі також є значним (за оцінками, до 30-40% витрат на пропускну здатність). Тому, хоча детальний аналіз пакетів може досягти високоточного виявлення аномалій, вимога зберігати та обробляти великі обсяги даних накладає значні технічні обмеження[6].

Окрім детального аналізу всіх пакетів, існують інші способи збору метаданих, які можуть зменшити навантаження на систему, а саме - NetFlow/sFlow.

1.3.2 Агреговані метрики потоків (NetFlow/sFlow)

Інтегрований метод збору показників, зокрема з використанням протоколів NetFlow і sFlow, призначений для запису метаданих мережевого потоку, включаючи базовий 5-кортеж (IP-адреси джерела та призначення, порти,

протоколи), тривалість сеансу та кількість переданих даних (кількість пакетів і байтів). Порівняно з PCAP ці підходи можуть значно зменшити навантаження на мережеву інфраструктуру, оскільки зберігаються лише зведені дані, зменшуючи використання сховища даних до 1-2% даних, згенерованих при використанні повного пакетного аналізу. Однак, хоча агрегація даних може ефективно зменшити навантаження, вона може призвести до втрати деяких деталей, важливих для поглибленого аналізу атаки. Таким чином, використання NetFlow/sFlow має знайти баланс між зменшенням витрат на ресурси та підтримкою достатньої цілісності інформації для виявлення аномалій.

На додаток до сукупних показників, дані, зібрані через журнали подій брандмауера, можуть бути корисними та можуть виявити конкретні спроби доступу та інші важливі події[7][8].

1.3.3 Логування подій брандмауерів

Використання журналу подій із системою брандмауера, наприклад використання iptables із цільовою дією LOG, дозволяє реєструвати спроби доступу, помилки автентифікації та інші важливі ознаки вашого мережевого середовища. Цей метод використовує записи системного журналу, які містять основні характеристики пакетів, такі як IP-адреси джерела та призначення, порти, інформацію про протокол і відповідні позначки TCP. Журнал подій дозволяє створювати кореляційні сценарії для виявлення зловмисної поведінки в реальному часі, дозволяючи швидко реагувати на загрози. При цьому ресурсоемність такого підходу мінімальна — для журналювання використовується менше 0,1% пропускної здатності, що робить його привабливим для впровадження в масштабованих системах моніторингу[7][9].

Таким чином, комбіноване використання PCAP, NetFlow/sFlow та iptables може створити багаторівневу систему моніторингу, яка закладає основу для більш детального формату та технічного аналізу, представленого в наступному розділі.

Метод	Характеристики			
	Обсяг даних	Вплив на мережу	Час обробки	Ефективність
PCAP Повний пакетний аналіз Детальний аналіз всіх пакетів	Високий ~1 ТБ/день	Високий 30-40%	Високий Складний	Дуже висока Детальний
NetFlow/sFlow Агреговані метрики потоків Збір метаданих мережевих потоків	Середній 1-2% від PCAP	Середній 5-10%	Середній Оптимальний	Висока Базові метрики
Журнали подій брандмауерів Логування системних подій і доступу	Низький <0.1% від PCAP	Дуже низький <1%	Низький Швидкий	Середня Цільові події

■ Високий рівень ■ Середній рівень ■ Низький рівень

Рисунок 1.3 – Порівняння методів збору оперативних даних

На рисунку 1.3 показано порівняння методів збору оперативних даних, що ілюструють ключові характеристики кожного методу: обсяг зібраних даних, вплив на пропускну здатність мережі, час обробки та ефективність у контексті виявлення аномалій. На малюнку видно, що PCAP надає повні дані з високою точністю, але в той же час займає багато ресурсів; NetFlow/sFlow дозволяє значно знизити навантаження, зберігаючи основні показники мережевого трафіку; Журнали подій брандмауера є ефективним інструментом для швидкого виявлення конкретних спроб доступу та аномалій з мінімальним споживанням ресурсів.

1.4 Методи аналізу мережевого трафіку

Спираючись на методи збору даних і системи виявлення вторгнень, описані в попередніх розділах, важливо розглянути методи аналізу мережевого трафіку для виявлення шкідливих дій у промислових мережах. Аналіз зібраних даних є ключовою частиною комплексної стратегії моніторингу, яка дозволяє виявляти

кіберзагрози та запобігати потенційним атакам на операційні та технічні системи. Для промислових мереж характерна необхідність забезпечення високої точності виявлення загроз і мінімального впливу на критичні виробничі процеси, що висуває особливі вимоги до вибору аналітичних методів.

На малюнку 1.4 наведено порівняння методів аналізу мережевого трафіку, ілюструючи ключові характеристики кожного: сигнатурний, поведінковий та гібридний. На малюнку показано принципи роботи різних методів, їх ефективність у виявленні відомих і невідомих загроз, а також їхні вимоги до обчислювальних ресурсів і точності виявлення аномалій у промислових мережевих середовищах [2][10].

	Сигнатурний	Поведінковий	Гібридний
Принцип	Пошук відомих шаблонів атак	Виявлення відхилень від нормальної поведінки	Комбінування обох методів
Ефективність	Відомі загрози: 95%+ Нові загрози: 15-30%	Відомі загрози: 75-85% Нові загрози: 70-90%	Відомі загрози: 95%+ Нові загрози: 75-95%
Ресурси	Низькі вимоги до ресурсів	Високі вимоги до ресурсів	Середньо-високі вимоги до ресурсів
Точність	FP: низькі (~5%) FN: високі (70-85%)	FP: високі (20-30%) FN: низькі (10-30%)	FP: середні (8-12%) FN: низькі (5-15%)
Особливості	+ Висока точність для відомих атак	+ Виявлення нових атак	+ Оптимальний баланс точності

FP - хибно-позитивні, FN - хибно-негативні результати

Рисунок 1.4 – Порівняння методів аналізу мережевого трафіку

1.4.1 Сигнатурний аналіз

Аналіз сигнатур базується на використанні попередньо визначених шаблонів (сигнатур), які характеризують відомі мережеві атаки та шкідливі дії. Метод передбачає порівняння мережевого трафіку з базою даних сигнатур для виявлення потенційних загроз. Ключовими компонентами аналізу сигнатур є формальні правила, які описують такі елементи, як послідовності пакетів, шаблони даних,

заголовки протоколів та інші характеристики мережевої активності, властиві певному типу атаки[11].

Ефективність сигнатурного підходу у виявленні відомих загроз дуже висока, з показником точності понад 95% для атак із доступними сигнатурами в базі даних (NIST SP 800-94). Це досягається шляхом чіткої формалізації сигналів зловмисної діяльності та зниження ймовірності помилкових спрацьовувань, що особливо важливо в промислових мережах, де несподівані зупинки процесів можуть призвести до значних фінансових втрат. Операційні витрати на підтримку аналізу підписів відносно низькі, оскільки такі системи не вимагають великої кількості обчислювальних ресурсів для обробки даних — у середньому системи на основі підписів використовують лише 10-15% ресурсів порівняно з поведінковими підходами[11].

Проте сигнатурний аналіз має суттєві обмеження, які становлять значний виклик для сучасних систем захисту промислових мереж[12]. Зокрема:

1. Нездатність ефективно ідентифікувати нові, невідомі загрози (уразливості нульового дня). Аналіз сигнатур може виявити лише атаки, для яких було створено сигнатури, створюючи вразливе часове вікно між появою нової атаки та оновленням бази даних сигнатур. Згідно з дослідженням (ICS-CERT), це вікно може коливатися від днів до тижнів, що є неприйнятним для критичної інфраструктури[12].
2. Проблема адаптації загальних сигнатур до специфіки протоколів ICS. Більшість комерційних баз даних функцій зосереджені на ІТ-середовищах і не враховують унікальні особливості промислових протоколів (Modbus, DNP3, Profinet). Тестування ефективності загальних сигнатур в середовищах ОТ показало, що точність виявлення конкретних атак на промислові системи впала до 60-75%[12].
3. Складність створення ефективних сигнатур для атак, що використовують легітимні команди промислових протоколів. Такі атаки, як маніпуляція уставками, можуть використовувати абсолютно коректні з точки зору

протоколу команди, що ускладнює їх виявлення за допомогою традиційних сигнатур[12].

4. Необхідність постійного оновлення бази даних сигнатур, що створює додаткове навантаження на адміністративний персонал та потенційно збільшує поверхню атаки через регулярне підключення до зовнішніх джерел оновлень[12].

Тестування ефективності сигнатурного підходу для виявлення нових загроз, таких як цілеспрямовані атаки на промислові системи (наприклад, Industroyer, TRITON), показало низьку ефективність — близько 15-30% для раніше невідомих векторів атак. Це тому, що зловмисник навмисно запускає атаку, яка обходить існуючі підписи або використовує нові вектори, для яких ще не створено підпис. Незважаючи на ці обмеження, сигнатурний аналіз залишається домінуючим підходом для виявлення зловмисної активності в промислових мережах через його надійність, низький рівень помилкових спрацьовувань і низькі вимоги до обчислювальних ресурсів. Однак його ефективність значною мірою залежить від якості та відповідності бази даних підписів і здатності адаптувати загальні правила до конкретних характеристик промислових протоколів і операційних технологій[12].

Хоча сигнатурний аналіз є високоефективним у виявленні відомих загроз, він має обмеження для нових атак, тому рекомендується розглянути альтернативні підходи, особливо поведінковий аналіз[11][12].

1.4.2 Поведінковий аналіз (anomaly-based detection)

Поведінковий аналіз, на відміну від сигнатурного, базується на створенні моделей нормальної активності мережі та виявленні відхилень від цих моделей. Даний підхід передбачає побудову статистичних або інтелектуальних моделей, що описують типову поведінку компонентів мережі, користувачів та протоколів, а потім ідентифікують аномалії, які можуть вказувати на наявність кіберзагроз. Ефективність поведінкового аналізу значною мірою залежить від точності

побудованих моделей та здатності системи розрізняти легітимні зміни в поведінці мережі від зловмисної активності[13][14].

Для реалізації поведінкового аналізу в промислових мережах використовуються різні методи машинного навчання, включаючи традиційні статистичні методи та алгоритми глибокого навчання. Зокрема:

- Статистичні методи, що вимірюють відхилення від нормального розподілу параметрів мережевого трафіку. Такі системи будують статистичні профілі на основі історичних даних і виявляють відхилення, що перевищують певні порогові значення[13][14].
- Методи кластеризації, які групують схожі події та виявляють аномалії як ізольовані точки або малі кластери, що значно відрізняються від основних груп[13][14].
- Нейронні мережі, особливо рекурентні архітектури та автоенкодери, здатні виявляти складні нелінійні залежності в даних та ідентифікувати аномальну активність[13][14].
- Алгоритми на основі правил асоціацій, які виявляють незвичні послідовності подій або комбінації параметрів, не характерні для звичайного функціонування промислових систем[13][14].

Основною перевагою поведінкового аналізу є можливість виявлення нових, раніше невідомих типів атак (0-днів), оскільки такі системи фокусуються на відхиленнях від нормальної поведінки, а не на наявності конкретних ознак шкідливої діяльності. У промислових мережевих середовищах, де системи часто працюють за передбачуваними шаблонами, поведінкова аналітика може бути високоефективною - до 85-90% ефективності у виявленні несанкціонованої активності [13][14].

Проте, поведінковий аналіз стикається з суттєвими викликами при застосуванні в OT/ICS-середовищах:

1. Висока частота хибно-позитивних спрацьовувань, яка за даними досліджень може сягати 20-30% у деяких реалізаціях, що є оскільки помилкове

блокування легітимної активності може призвести до зупинки виробничих процесів [14].

2. Складність побудови точних моделей нормальної поведінки в гетерогенному середовищі промислових мереж, де використовуються різні протоколи та обладнання різних виробників з унікальними характеристиками трафіку [14].
3. Високі вимоги до обчислювальних ресурсів, особливо для методів глибокого навчання, що може бути проблематичним для систем з обмеженою потужністю, які часто зустрічаються в промислових середовищах. Системи на основі поведінкового аналізу можуть потребувати в 5-10 разів більше обчислювальних ресурсів порівняно з сигнатурними підходами [14].
4. Необхідність тривалого періоду навчання для створення точних моделей нормальної поведінки, що може становити від кількох тижнів до кількох місяців для складних промислових систем. Протягом цього періоду система залишається вразливою до атак, оскільки моделі ще не досягли оптимальної точності [14].
5. Труднощі з адаптацією до легітимних змін у поведінці мережі, таких як планові оновлення, зміни конфігурації або сезонні коливання виробничих процесів, що можуть викликати хибні тривоги [14].

Незважаючи на ці обмеження, поведінковий аналіз залишається важливим компонентом сучасних систем захисту промислових мереж, особливо для виявлення складних цілеспрямованих атак, які неможливо визначити за допомогою традиційних сигнатур. Поведінкова аналітика особливо ефективна в середовищах, де операції дуже передбачувані, наприклад, на автоматизованих виробничих лініях або системах керування процесами, де будь-яке відхилення від нормальних шаблонів може свідчити про зловмисну діяльність [13][14].

Враховуючи сильні сторони та обмеження сигнатурного та поведінкового підходів, розглянемо гібридні підходи, які компенсують їхні відповідні недоліки..

1.4.3 Гібридні підходи до виявлення загроз

Гібридні підходи до аналізу мережевого трафіку поєднують сигнатурні та поведінкові методи, щоб подолати обмеження кожного підходу та підвищити загальну ефективність виявлення загроз. Такі системи поєднують бази даних сигнатур відомих атак із моделями нормальної поведінки мережі, що робить їх ефективними у виявленні відомих загроз із низьким рівнем помилкових спрацьовувань і здатними ідентифікувати нові, раніше невідомі вектори атак на основі аномальної активності [1][15].

У контексті промислових мереж гібридні підходи реалізуються за такими основними принципами:

1. Багаторівнева архітектура аналізу, де сигнатурний аналіз застосовується як перший рівень фільтрації для виявлення відомих загроз, а поведінковий аналіз використовується для більш глибокого дослідження потенційно підозрілої активності, яка не була ідентифікована за допомогою сигнатур [15].
2. Інтеграція результатів різних методів аналізу з використанням кореляційних алгоритмів, що дозволяє знизити рівень хибно-позитивних спрацьовувань поведінкового аналізу. За даними досліджень, гібридні підходи дозволяють знизити кількість хибних тривог на 40-60% порівняно з використанням виключно поведінкового аналізу [16].
3. Застосування контекстної інформації та знань про специфіку промислових систем для підвищення точності виявлення загроз. Зокрема, врахування таких факторів, як робочі цикли виробничих процесів, планові технічні роботи та специфіка промислових протоколів, дозволяє значно підвищити ефективність аналізу [1].
4. Використання механізмів зворотнього зв'язку, коли результати поведінкового аналізу використовуються для автоматичного генерування нових сигнатур, що дозволяє оперативно реагувати на виявлені нові типи загроз та інтегрувати їх у базу знань системи [15].

Емпіричні дані підтверджують ефективність гібридних підходів у промислових мережах: використання комбінованого підходу може підвищити точність виявлення загроз на 15-20% порівняно із застосуванням лише сигнатур або аналізу поведінки (ІЕС 62443). Підвищення ефективності є особливо значним при виявленні складних, багатоетапних атак, які використовують як відомі вектори, так і нові методи для обходу традиційних систем захисту[1].

Проте, гібридні системи також стикаються з певними викликами, зокрема:

1. Підвищена складність реалізації та налаштування, що вимагає високої кваліфікації адміністративного персоналу та детального розуміння як загальних принципів кібербезпеки, так і специфіки промислових систем [15].
2. Високі вимоги до обчислювальних ресурсів, особливо при застосуванні складних алгоритмів кореляції та машинного навчання в режимі реального часу. За оцінками експертів, гібридні системи можуть потребувати в 1,5-2 рази більше ресурсів порівняно з сигнатурними підходами [17].
3. Складність інтеграції з існуючими системами захисту та моніторингу, особливо в гетерогенних середовищах з обладнанням різних виробників та різноманітними протоколами [15].
4. Необхідність постійного оновлення баз сигнатур, та моделей нормальної поведінки мережі, що вимагає значних адміністративних та обчислювальних ресурсів [15].

Незважаючи на зазначені проблеми, гібридні підходи швидко набувають популярності в промислових середовищах завдяки своїй здатності забезпечувати всебічний захист від широкого спектру загроз. Гібридні системи особливо ефективні в критичній інфраструктурі, де баланс між високою точністю виявлення загроз і низьким рівнем помилкових тривог є вирішальним [1][15].

Аналізуючи існуючі методи виявлення загроз промислових мереж, можна зробити висновок, що кожен метод має свої переваги та обмеження. Однак розробка спеціальних правил підпису, адаптованих до характеристик промислових протоколів і в поєднанні з елементами поведінкового аналізу, є одним із

найперспективніших напрямків підвищення рівня захисту середовищ OT/ICS від сучасних кіберзагроз.

1.5 Спеціалізовані рішення для моніторингу промислових протоколів

Ефективність моніторингу та захисту промислових мереж значною мірою залежить від здатності системи безпеки аналізувати конкретні промислові протоколи, які суттєво відрізняються від традиційних IT-протоколів. Промислові протоколи були розроблені в епоху, коли операційні та технічні системи працювали в ізольованих середовищах, тому багатьом з них бракує вбудованих механізмів безпеки, таких як шифрування, автентифікація та контроль цілісності даних. У комбінації OT та IT-систем, описаних у попередніх розділах, ці вразливості становлять значний ризик для критичної інфраструктури та вимагають спеціальної стратегії моніторингу [4][5].

У багаторівневій моделі Університету Пердью, розглянутій у розділі 1.1, промислові протоколи зазвичай працюють на рівнях 0-2, забезпечуючи зв'язок між фізичними пристроями, локальними контролерами та системами SCADA/HMI. Впровадження спеціалізованого рішення для моніторингу таких протоколів є ключовою частиною загальної сегментації мережі та оперативної стратегії збору даних, описаної в попередніх розділах.

Порівняння методів моніторингу промислових протоколів

Протокол	Характеристики	Вразливості	Методи моніторингу	Ефективність
Modbus (промисловий стандарт)	- Простий - Ведучий-підлеглий - Функціональні коди	- Немає автентифікації - Немає шифрування - Неавторизований запис	- Аналіз функц. кодів - Контроль частоти - Перевірка реєстрів	85-90% для відомих 40-50% для нових
DNP3 (енергетичні системи)	- Складніша архітектура - Багатостанційний - Звітвання по подіях	- Базова версія незахищена - Вразливість команд - Підміна команд	- Контроль послідовності - Аналіз часових інтервалів - DNP3-SA автентифікація	80-85% стандартна 90-95% з DNP3-SA
Profinet (виробничі лінії)	- Ethernet-базований - Реальний час (<1 мс) - Різні профілі пристроїв	- Складна імплементація - Атаки типу "людина посередині"	- Аналіз часових моделей - Моніторинг циклів - Контроль конфігурації	75-85% для відомих 30-40% для складних
OPC UA (сучасна інтеграція)	- Вбудовані механізми безпеки - Шифрування - Сервіс-орієнтований	- Складне налаштування - Зашифрований трафік не видно для аналізу	- Аналіз сесій - Контроль підписок - Логи безпеки серверів	95-97% для відомих типів атак

Ефективність - відсоток виявлених аномалій і атак

Рисунок 1.5 – Порівняння методів моніторингу промислових протоколів

На рисунку 1.5 показано порівняння методів моніторингу промислових протоколів, що ілюструє характеристики кожного основного промислового протоколу (Modbus, DNP3, Profinet, OPC UA), їхні слабкі сторони та ефективність різних методів моніторингу для кожного протоколу. Цифра також відображає відносну складність реалізації та ресурсоємність кожного підходу моніторингу в реальних промислових умовах [4][5].

1.5.1 Моніторинг специфічних промислових протоколів

Моніторинг певного промислового протоколу вимагає глибокого розуміння його структури, функціональних характеристик і загальних шаблонів зв'язку. Розглянемо основні протоколи та стратегії, які він контролює:

Modbus є одним із найстаріших і найпоширеніших промислових протоколів, що працює за принципом «ведучий-підлеглий» і використовується для взаємодії з ПЛК і RTU. Структура протоколу Modbus є відносно простою, з функціональними кодами для зареєстрованих операцій читання/запису та дискретного введення/виведення. Основним недоліком Modbus є відсутність механізмів

автентифікації та шифрування, що дозволяє зловмисникам виконувати неавторизовані операції запису та читання даних (ІЕС 62443-3-3). Згідно з дослідженнями, понад 75% атак на системи Modbus використовують цю вразливість для маніпулювання ключовими параметрами[1][2].

Моніторинг Modbus зосереджений на виявленні ненормальних послідовностей функціонального коду та нетипових операцій. Спеціалізовані системи моніторингу Modbus аналізують:

- Частоту та послідовність запитів
- Діапазони адрес реєстрів для операцій читання/запису
- Значення, що записуються в критичні реєстри
- Запити до неіснуючих пристроїв або функціональні коди, не підтримувані цільовими пристроями

Ефективність таких систем для виявлення аномалій у Modbus-трафіку становить 85-90% для відомих типів атак, проте значно знижується (до 40-50%) для складних цілеспрямованих атак, які імітують легітимні послідовності команд (NIST SP 800-82r2) [1][2].

DNP3 (протокол розподіленої мережі) є більш складним протоколом, який використовується в основному в системах живлення, забезпечуючи надійну передачу даних і можливість визначати пріоритети повідомлень. На відміну від Modbus, DNP3 підтримує модель запиту-відповіді з кількома станціями, що дозволяє ведучому зв'язуватися з кількома підлеглими. DNP3 пропонує вищий рівень складності з розширеними функціями, такими як звітність на основі подій і відмітка часу, але базова версія також має обмежені механізми безпеки [1][2].

Системи моніторингу DNP3 аналізують:

- Функціональні коди та об'єкти у запитах/відповідях
- Послідовності команд та їх відповідність операційним шаблонам
- Часові інтервали між запитами та аномальні затримки
- Неавторизовані операції, особливо прямі операції керування

Ефективність моніторингу DNP3 коливається в межах 80-85% для виявлення аномальної активності, проте наявність розширення DNP3-SA (Secure

Authentication) підвищує цей показник до 90-95% за рахунок додаткової аутентифікації критичних операцій [2].

Profinet є високопродуктивним промисловим протоколом на базі Ethernet, що використовується для комунікації в режимі реального часу, особливо в автоматизованих виробничих середовищах. Profinet забезпечує передачу даних з затримками менше 1 мс, що є критичним для систем керування рухом та синхронізованих операцій.

Моніторинг Profinet є складнішим через різноманітність імплементацій та профілів пристроїв, але зосереджується на:

- Аналізі часових моделей обміну даними
- Відстеженні відхилень від очікуваних циклів зв'язку
- Моніторингу зміни конфігурації пристроїв та топології мережі
- Виявленні аномальних пакетів, що можуть вказувати на атаки типу "людина посередині"

Ефективність моніторингу Profinet значною мірою залежить від якості реалізації та знання конкретного середовища, з коефіцієнтом виявлення 75-85% для відомих типів атак, але менш ефективний (близько 30-40%) при виявленні складних атак, спрямованих на маніпулювання циклом зв'язку [2].

OPC UA (OPC Unified Architecture) — сучасний платформно-незалежний стандарт обміну даними між пристроями та системами керування. На відміну від старих протоколів, OPC UA має вбудовані механізми безпеки, включаючи шифрування, автентифікацію та авторизацію. Це робить OPC UA більш стійким до традиційних атак, але створює нові проблеми для моніторингу через обмежену видимість зашифрованого трафіку.

Системи моніторингу OPC UA зосереджуються на:

- Аналізі встановлення сесій та процедур аутентифікації
- Моніторингу шаблонів доступу до даних та сервісів
- Відстеженні аномальних змін у передплатах на дані
- Виявленні неавторизованих змін у конфігурації безпеки

Ефективність моніторингу OPC UA є найвищою серед промислових протоколів – до 95-97% для відомих типів атак завдяки вбудованим механізмам безпеки. Проте, зашифрований характер трафіку створює "чорну скриньку" для традиційних систем моніторингу, що вимагає інтеграції з серверами OPC UA для отримання детальних логів безпеки[2].

Загальними обмеженнями існуючих рішень для моніторингу промислових протоколів є:

1. Недостатня адаптація до специфіки виробничих середовищ, що вимагає тривалого налаштування та створення індивідуальних правил моніторингу для кожної системи [1][2].
2. Висока частота хибно-позитивних спрацьовувань, особливо в гетерогенних середовищах з обладнанням різних виробників, що може призвести до ігнорування попереджень персоналом [2].
3. Складність ідентифікації складних багатоетапних атак, які використовують легітимні команди та операції для досягнення зловмисної мети [2].
4. Відсутність стандартизованих підходів до моніторингу, що призводить до фрагментації рішень та складностей при інтеграції з існуючими системами безпеки.
5. Неєфективність традиційних методів для виявлення атак, що використовують особливості протоколів, наприклад, маніпуляції з уставками в межах дозволених діапазонів, але з потенційно небезпечними наслідками.

Хоча моніторинг специфічних характеристик промислових протоколів дозволяє виявляти певні типи аномалій, більш детальний аналіз вмісту пакетів за допомогою deep packet inspection забезпечує вищий рівень захисту від складних атак.

1.5.2 Deep packet inspection для промислових протоколів

Глибока перевірка пакетів (DPI) — це комплексний метод аналізу мережевого трафіку, який передбачає детальний аналіз заголовків пакетів і їх корисного навантаження. На відміну від традиційних методів моніторингу, які зосереджені на аналізі метаданих потоку, DPI дозволяє розшифровувати та

перевіряти фактичний вміст пакетів, включаючи команди, параметри та дані, надіслані промисловими протоколами [18][19].

Принципи роботи DPI в контексті промислових протоколів включають:

1. Глибокий синтаксичний та семантичний аналіз пакетів з урахуванням специфіки кожного протоколу. DPI-системи інтерпретують дані відповідно до структури та правил конкретного промислового протоколу, що дозволяє розуміти контекст та зміст команд [20].
2. Виявлення аномалій на рівні корисного навантаження, включаючи нетипові значення параметрів, послідовності команд та шаблони даних, які можуть вказувати на зловмисну активність[19].
3. Перевірка відповідності команд та операцій встановленій моделі нормальної поведінки системи, що базується на історичних даних та заданих політиках безпеки [21].
4. Кореляція інформації з різних пакетів та потоків для виявлення розподілених та багатоетапних атак, які можуть залишатися непоміченими при поверхневому аналізі [19].

Ефективність DPI для промислових протоколів була підтверджена кількісними даними: дослідження показали, що впровадження DPI може збільшити рівень виявлення атак на промислові системи на 25-30% порівняно з традиційними методами моніторингу (ENISA, 2021). Це значно підвищує ефективність під час виявлення складних атак, наприклад тих, де маніпуляції налаштуваннями та командами спричиняють непередбачувану взаємодію між компонентами системи[18].

Однак, впровадження DPI в ОТ-середовищах стикається з низкою технічних обмежень та викликів:

1. Вплив на продуктивність мережі. Глибокий аналіз пакетів вимагає значних обчислювальних ресурсів та може створювати затримки при передачі даних. В промислових системах, де затримки можуть бути критичними, DPI має бути ретельно оптимізованим. За вимірюваннями, затримка, що вноситься DPI-системами, коливається в межах 1-5 мс, що може бути неприйнятним для

деяких застосувань реального часу з високими вимогами до швидкодії, таких як системи керування рухом з вимогами затримки менше 1 мс [19].

2. Обмежена видимість зашифрованого трафіку. Для протоколів, що використовують шифрування (OPC UA, деякі імплементації Profinet з TLS), DPI має обмежену ефективність без доступу до ключів дешифрування, що створює додаткові виклики для моніторингу безпеки [20].
3. Складність інтеграції з існуючими ОТ-системами. Впровадження DPI часто вимагає внесення змін до мережевої архітектури, що може бути проблематичним для критичних систем з високими вимогами до доступності та надійності [20].
4. Високі вимоги до знань та експертизи. Налаштування та підтримка DPI-систем для промислових протоколів вимагає глибокого розуміння як специфіки цих протоколів, так і особливостей конкретних виробничих процесів [21].
5. Проблеми з масштабованістю. При збільшенні розміру мережі та кількості пристроїв зростає обсяг трафіку, що має бути проаналізований, що призводить до експоненційного зростання вимог до обчислювальних ресурсів DPI-систем.

Незважаючи на ці виклики, DPI залишається одним з найефективніших методів моніторингу промислових протоколів, особливо у поєднанні з іншими підходами, такими як аналіз метаданих потоків та поведінковий аналіз. Успішні приклади застосування DPI для виявлення атак на промислові системи включають:

- Виявлення атак на електроенергетичні системи, що використовують маніпуляції з уставками в DNP3-протоколі для створення нестабільності в мережі (ефективність виявлення до 92%) [20].
- Ідентифікація зловмисних операцій запису в реєстри пристроїв Modbus, які можуть призвести до фізичного пошкодження обладнання (ефективність виявлення до 88%) [18].

- Виявлення man-in-the-middle атак на Profinet-трафік, що використовуються для підміни команд керування або викрадення даних (ефективність виявлення до 85%) [19].

Однак, щоб подолати обмеження DPI та підвищити загальну ефективність захисту промислової мережі, необхідно розробити інтегровані рішення, які поєднують сильні сторони різних методів моніторингу та аналізу трафіку. Ці рішення повинні враховувати деталі промислових протоколів, характеристики виробничих процесів і баланс між безпекою та функціональністю критичних систем.

Розробка спеціальних методів моніторингу та захисту для кожного промислового протоколу є необхідною умовою для забезпечення кібербезпеки в сучасних середовищах OT/ICS. Поєднання глибокого розуміння деталей протоколу з сучасними методами аналізу даних забезпечує основу для ефективного виявлення та запобігання кіберзагрозам у промислових мережах.

1.6 Критичний аналіз обмежень існуючих методів моніторингу промислових мереж

Незважаючи на значний прогрес у розробці методів моніторингу та аналізу мережевого трафіку в промислових системах, існуючі підходи все ще мають кілька істотних обмежень, які знижують їхню ефективність у сучасних середовищах OT/ICS. Аналіз літератури та практичного досвіду впровадження систем моніторингу дозволяє виявити критичні недоліки, які необхідно комплексно усунути для забезпечення належного рівня кібербезпеки промислових об'єктів [1, 18, 19].

1.6.1 Ресурсні обмеження повного пакетного аналізу в умовах промислових мереж

Хоча підхід повного аналізу пакетів (PCAP) забезпечує найвищий рівень деталізації під час аналізу мережевого трафіку, він створює надмірне навантаження на інфраструктуру систем OT/ICS, що є ключовим обмеженням для його широкого

використання. Повний аналіз пакетів промислової мережі середнього розміру може генерувати від 5 до 12 ТБ даних на місяць, що вимагає значних ресурсів для зберігання та обробки цієї інформації. Для ефективного аналізу таких великих обсягів даних у режимі реального часу необхідна обчислювальна потужність повинна в 3-5 разів перевищувати типові можливості мережевої інфраструктури промислових об'єктів [19][21].

Кількісний аналіз ресурсних витрат PCAP показує, що для мережі, що обслуговує 100 промислових пристроїв, з середньою пропускнуою здатністю 100 Мбіт/с, необхідно забезпечити:

1. Виділене сховище даних обсягом від 150 до 360 ТБ для зберігання річного архіву PCAP-даних, вартість якого становить від \$15,000 до \$50,000, залежно від обраного рішення та вимог до надійності [20].
2. Високопродуктивні сервери аналізу даних із загальною вартістю від \$25,000 до \$75,000, здатні обробляти потоковий трафік без створення затримок у мережі [21].
3. Додаткову мережеву інфраструктуру для дзеркалювання трафіку, що може становити від 10% до 30% вартості всього мережевого обладнання промислового об'єкта [21].

Такі витрати є економічно необґрунтованими для багатьох підприємств, особливо середнього та малого масштабу, згідно з яким лише 15-20% промислових підприємств можуть дозволити собі повноцінне впровадження PCAP-аналізу на всіх рівнях моделі Purdue [12][19].

Окрім фінансових аспектів, аналіз PCAP також може стати значним тягарем для мережевої інфраструктури. Згідно з вимірюваннями ICS-CERT, впровадження моніторингу PCAP збільшує загальне навантаження на мережу на 30-40%, що є неприйнятним для систем реального часу з високою пропускнуою здатністю та низькими вимогами до затримки. Цей вплив на продуктивність мережі прямо суперечить фундаментальним вимогам до промислових систем, описаним у розділі 1.1, де затримка передавання даних є критичною для забезпечення безпечної та стабільної роботи процесу.

Розрахунки, проведені NIST (SP 800-82r3, 2023), показують, що для промислових протоколів з високими вимогами до продуктивності, таких як EtherCAT або PROFILT, прийнятна затримка становить менше 1 мс, а реалізація повного аналізу пакетів збільшить затримку на 5-15%, що перевищує допустимі параметри для систем керування рухом та інших критичних програм [2][20][21].

1.6.2 Проблеми доступності та адаптивності спеціалізованих рішень

Ринок спеціалізованих засобів моніторингу промислових мереж характеризується високою концентрацією пропрієтарних рішень, що створює суттєву перешкоду для їх широкого впровадження в різноманітних середовищах OT/ICS. За даними Gartner (2023), понад 75% поточних комерційних систем моніторингу промислових мереж є закритими рішеннями з обмеженими можливостями налаштування та інтеграції з існуючою інфраструктурою. За даними Forrester Research (2022), для типового підприємства така система коштуватиме від 50 000 до 500 000 доларів США, що робить її недоступною для значної частини потенційних користувачів [19][20].

Ключовим обмеженням є відсутність достатньої кількості зрілих рішень з відкритим кодом для моніторингу та аналізу трафіку в середовищах OT/ICS. Аналіз, проведений Cyber X (2023), показує, що лише близько 10-15% функціональності комерційної системи можна використовувати у відкритих рішеннях без подальшого розвитку та адаптації. Це призводить до того, що організації з обмеженим бюджетом змушені відмовлятися від впровадження систем моніторингу або використовувати рішення, які не оптимізовані для промислових середовищ, що потенційно знижує їхню ефективність. Проблема ускладнюється специфікою промислових протоколів, розглянутих раніше. Згідно з дослідженням ISA (2023), для ефективного моніторингу в промислових середовищах система повинна підтримувати аналіз щонайменше 15-20 різних протоколів, включаючи їх варіанти та версії. Більшість доступних рішень з відкритим вихідним кодом забезпечують лише повну підтримку 3-5 найпоширеніших протоколів, залишаючи

велику частину конкретного промислового трафіку без належного аналізу [11][3][21].

Огляд сучасних open-source проєктів у сфері моніторингу промислових мереж, проведений SANS Institute (2023), виявив наступні кількісні показники їхніх обмежень [14][21]:

1. Підтримка промислових протоколів: 25-30% від необхідного для комплексного аналізу промислових мереж.
2. Можливості виявлення специфічних для ОТ загроз: 40-45% порівняно з комерційними аналогами.
3. Інтеграція з системами автоматизованого реагування: 15-20% від функціональності, доступної в пропрієтарних рішеннях.
4. Підтримка специфічних для конкретних галузей промисловості вимог та сценаріїв: менше 10%.

Ці обмеження суттєво ускладнюють створення ефективних систем моніторингу на базі відкритих рішень без значних інвестицій у їхню доробку та адаптацію, що часто нівелює економічні переваги використання open-source програмного забезпечення.

1.6.3 Недоліки існуючих підходів до аналізу аномалій у промисловому трафіку

Аналіз підходів до виявлення аномалій, розглянутих раніше, показує, що вони недостатньо адаптовані до конкретного випадку промислових мереж, де шаблони трафіку, як правило, мають високі ймовірності переходу та періодичні, і де концепція «нормальної поведінки» суттєво відрізняється від концепції в IT-середовищах[21].

За даними Claroty Research (2023), традиційні методи виявлення аномалій мають такі недоліки при застосуванні до систем OT/ICS[14]:

1. Висока частота хибно-позитивних спрацьовувань, що за даними спостережень у реальних промислових мережах досягає 30-40% для систем на основі загальних статистичних методів. Це призводить до "втоми від

сповіщень" (alert fatigue) серед операторів та потенційного ігнорування реальних інцидентів[21].

2. Недостатня ефективність виявлення специфічних для промислових систем атак, зокрема тих, що використовують легітимні команди з аномальними параметрами. За вимірюваннями в контрольованому середовищі, точність виявлення таких атак не перевищує 50-60% при використанні загальних методів аналізу аномалій[21].
3. Обмежена здатність до адаптації до змін у нормальній поведінці системи, пов'язаних з плановими операціями, такими як технічне обслуговування, калібрування або зміни виробничих циклів. Дослідження показують, що до 65% хибних спрацьовувань пов'язані саме з плановими змінами в операційних процесах[21].

Найбільш відчутною ця проблема стає при застосуванні методів сигнатурного аналізу, що були згадані раніше. Згідно зі статистикою ICS-CERT, від 70 до 80 відсотків сигнатур, розроблених для IT-інфраструктур, демонструють низьку ефективність в промислових мережах без суттєвого налаштування. Це вимагає значних ресурсів та експертних знань. Водночас, опитування SANS (2023) показує, що лише чверть або третина організацій мають співробітників з необхідною кваліфікацією для такого пристосування.

1.6.4 Економічні та операційні обмеження

Впровадження систем моніторингу та аналізу трафіку в промислових мережах стикається зі значними фінансовими та операційними обмеженнями, що суттєво звужує коло організацій, здатних розгорнути комплексні рішення. Згідно з дослідженням Інституту Понемона (2023), загальна вартість володіння (ТСО) комерційних систем моніторингу промислових мереж протягом п'ятирічного періоду становить[21]:

1. Для малих підприємств (до 100 промислових пристроїв): \$150,000-350,000
2. Для середніх підприємств (100-500 пристроїв): \$350,000-900,000
3. Для великих підприємств (понад 500 пристроїв): \$900,000-3,000,000+

Ці інвестиції є непропорційними загальним бюджетам кібербезпеки, особливо для малих та середніх підприємств. Згідно з опитуванням ICS Security Survey (2023), середній бюджет кібербезпеки малих та середніх промислових підприємств становить 50 000–250 000 доларів на рік, що робить повномасштабне впровадження сучасних систем моніторингу економічно недоцільним.

Функціональні обмеження включають[20]:

Потреба у висококваліфікованому персоналі для налаштування та підтримки систем моніторингу. Дослідження ISA показують, що для ефективної роботи системи моніторингу середньої промислової мережі потрібно щонайменше 2–3 експерти з глибокими знаннями як промислових технологій, так і кібербезпеки.

1. Необхідність у високкваліфікованому персоналі для налаштування та підтримки систем моніторингу. За дослідженнями ISA, для ефективної експлуатації системи моніторингу промислової мережі середнього масштабу необхідно щонайменше 2-3 спеціалісти з глибокими знаннями як промислових технологій, так і кібербезпеки [14][19].
2. Складність інтеграції з наявними процесами та системами. Згідно з даними ARC Advisory Group, до 60% проєктів з впровадження систем моніторингу промислових мереж перевищують заплановані терміни та бюджети через непередбачувані складнощі інтеграції [20].
3. Потенційний негативний вплив на продуктивність основних систем. Дослідження Gartner показують, що до 25% організацій відчували тимчасові збої в роботі певних компонентів інфраструктури ОТ під час впровадження систем моніторингу [21].

Ці економічні та операційні обмеження створюють значний розрив між теоретично можливим рівнем захисту та тим, який практично досяжний для більшості промислових підприємств, підкреслюючи необхідність розробки більш доступних та менш ресурсоємних рішень.

1.6.5 Необхідність розвитку альтернативних підходів

Критичний аналіз поточних методів моніторингу промислових мереж вказує на необхідність розробки альтернативних підходів, які поєднують достатню ефективність виявлення загроз з прийнятними вимогами до ресурсів та можливістю адаптації до специфічних середовищ OT/ICS.

Перспективним напрямом є розробка гібридних рішень, що інтегрують агреговані метрики трафіку (наприклад, NetFlow/sFlow), які розглянуто в розділі 1.4.2, з адаптивними методами виявлення аномалій та оптимізованими правилами сигнатур для конкретних промислових атак. Згідно з розрахунками Frost & Sullivan [21], такий підхід може забезпечити:

1. Зменшення потреби у зберіганні даних на 85-95% порівняно з повним аналізом пакетів, зберігаючи при цьому здатність виявляти до 80-85% відповідних загроз.
2. Зменшення навантаження на мережеву інфраструктуру на 70-80%, що критично важливо для систем з високими вимогами до продуктивності.
3. Зменшення сукупної вартості володіння на 60-70% порівняно з комерційними рішеннями на основі PCAP, що робить такі системи доступними для ширшого кола підприємств.

Дослідження, проведені Інститутом SANS (2023), підтверджують, що поєднання аналізу метаданих трафіку з цілеспрямованими перевірками вмісту пакетів лише для підозрілого трафіку дозволяє досягти оптимального балансу між ефективністю виявлення загроз та споживанням ресурсів [21].

Розробка рішень з відкритим кодом, адаптованих до промислових середовищ, є особливо перспективною. Ці рішення можна гнучко налаштовувати для задоволення конкретних потреб окремих галузей та організацій. Аналіз тенденцій, проведений ICS-CERT, вказує на поступове збільшення кількості та якості відкритих інструментів для моніторингу промислових мереж, хоча їх функціональність поки що поступається комерційним аналогам [12].

Підрахунки, пов'язані з ефективністю віртуального рішення, що базується на NetFlow та поглибленому аналізі, націленому на підозрілий трафік, демонструють можливість виявлення [21]:

- 1 До 95% атак, пов'язаних з аномальними обсягами або шаблонами трафіку (DoS, сканування, широкомовлення).
- 2 До 85% атак, що базуються на нестандартних з'єднаннях, послідовностях команд.
- 3 До 75% складних, цілеспрямованих атак на промислові системи.

Водночас, вимоги до ресурсів для такого рішення становлять лише 10-15% від повноцінної системи аналізу PCAP з аналогічною ефективністю, що робить його економічно доцільним для значно більшої кількості організацій.

ВИСНОВКИ ДО РОЗДІЛУ 1

Аналіз обмежень поточних методів моніторингу промислових мереж виявляє суттєві недоліки, що стримують їх широке впровадження та ефективне використання в різноманітних середовищах OT/ICS. Основні проблеми включають надмірне споживання ресурсів методами повнопакетного аналізу, високу вартість та закритий характер комерційних рішень, недостатню адаптацію методів аналізу до специфіки промислових мереж, а також значні економічні та операційні бар'єри для впровадження.

Ці обмеження створюють великий розрив між теоретичним та практичним рівнем захисту, який може бути досягнутий для більшості промислових компаній, що підкреслює необхідність розробки альтернативних підходів. Перспективним напрямком є розробка гібридних рішень, що базуються на аналізі метаданих потоків з цілеспрямованими поглибленими перевірками підозрілого трафіку, а також створення адаптованих інструментів з відкритим вихідним кодом, здатних збалансувати ефективність виявлення загроз та прийнятні вимоги до ресурсів.

Такі альтернативні підходи, у поєднанні з покращеними сигнатурними правилами, специфічними для промислового середовища, можуть забезпечити відповідний рівень захисту з набагато меншими вимогами до ресурсів, роблячи ефективний моніторинг промислових мереж доступним для значно ширшого кола організацій.

2 МОДЕЛЬ ВИЯВЛЕННЯ ТРАФІКУ В ICS-СЕРЕДОВИЩАХ

2.1 Опис моделі виявлення трафіку в ICS-СЕРЕДОВИЩАХ

Аналіз деталей середовища OT/ICS у розділі 1 виявив кілька унікальних характеристик промислових мереж, які суттєво відрізняють їх від традиційних ІТ-систем. Враховуючи ці характеристики, ми пропонуємо принципово інший підхід до виявлення аномалій мережевої активності, заснований на виявленні однорангових користувачів замість традиційної багаторівневої системи оцінки.

2.1.1 Архітектурна модель системи

Архітектура системи складається з чотирьох шарів :

Таблиця 2.1 – Архітектура системи

№	Шар	Функція	Принцип обробки
1	Edge-агент	локальний z-score, прості порогові правила	peer/edge
2	NetFlow Collector	ETL pipeline ($T_1 + T_2$)	hub-and-spoke
3	Suricata IPS	сигнатурне rule-matching	автономне (per-flow)
4	Core Correlator	кореляція $\phi(D,R,H)$ та створення інцидентів	централізовано

Тепер розглянемо кожну частину архітектури детальніше

Точка збору даних NetFlow. Фізичний або віртуальний мережевий пристрій (маршрутизатор, комутатор зі SPAN-портом, sFlow/NetFlow-пробірник), на якому активовано експорт записів NetFlow (v9/IPFIX).

Ці точки розташовані на ключових вузлах мережевої інфраструктури відповідно до ієрархії моделі Purdue. Оптимальне розміщення визначається топологічною моделлю мережі, яка формально може бути представлена у вигляді графа.

$$G(V, E), (2.1)$$

де V – множина вузлів мережі, а E – множина їх з'єднань.

В одноранговій архітектурі ці точки функціонують як автономні вузли первинного виявлення аномалій, що забезпечує децентралізацію та усуває єдину точку відмови.

Колектор NetFlow - центральний вузол, що приймає, нормалізує та збагачує записи, не беручи участі у прийнятті рішень про аномалії.

Колектор забезпечує централізований збір і базову обробку даних мережевого потоку. ETL-перетворення моделюється композицією функцій.

$$T_1: D \rightarrow D^n (\text{нормалізація}), T_2: D^n \rightarrow D' (\text{збагачення}), D' = T_2 \circ T_1(D) \quad (2.2),$$

- T_1 – перетворення *raw UDP-export* $\rightarrow$ *IPFIX v10*.
- T_2 – додавання *GeoIP*-міток, номера рівня *Purdue*, тощо
- D – множина необроблених даних, отриманих з точок збору,
- D' – нормалізована множина даних, підготовлена для аналізу.

Тому хоча колектор є централізованим елементом, він не порушує принципу одноранговості, оскільки слугує лише для агрегації і не бере участі в прийнятті рішень про наявність аномалій.

1. **Модуль аналізу.** Цей модуль виконує статистичну обробку та визначає відхилення від заданих базових показників. Робота цього модуля базується на математичній моделі, описаній у розділі 2.2. У контексті однорангової архітектури модуль аналізу забезпечує уніфіковану обробку даних від усіх рівноправних вузлів моніторингу.
2. **Suricata IDS/IPS.** Система інтегрує спеціальні правила для виявлення аномалій, унікальних для промислових мереж. Він працює за принципом кінцевих автоматів, де кожне правило представляє перехід між станами на основі вхідних даних. Реалізує парадигму однорангової обробки через автономне застосування правил до кожного потоку даних без ієрархічного підпорядкування.
3. **Система координації та інтеграції діяльності.** Система поєднує дані з різних джерел для створення повної картини ситуації з безпекою. У математиці це робиться шляхом інтегрування функції

$$A: \{e_1, e_2, \dots, e_n\} \rightarrow E, \quad (2.3)$$

де e_i – окремі події, а E – агрегована подія вищого рівня.

Забезпечує консолідацію результатів від автономних вузлів виявлення без порушення їх рівноправності в процесі аналізу.

Запропонована однорангова архітектура має наступні ключові характеристики:

1. **Децентралізація процесу виявлення** – кожна точка збору даних NetFlow функціонує як автономний детектор аномалій з власними механізмами аналізу.
2. **Рівноправність вузлів** – всі точки моніторингу мають однаковий статус у процесі виявлення аномалій, без ієрархічного підпорядкування.
3. **Незалежне прийняття рішень** – рішення про наявність аномалії приймається на рівні окремого вузла.
4. **Колективна обробка подій** – незважаючи на автономність у виявленні, система використовує централізований механізм кореляції та агрегації для формування комплексного представлення про безпеку мережі.

2.1.2 Модель обробки даних

Процес обробки даних поділяється на кілька послідовних етапів.:

1. **Збір даних NetFlow з точок моніторингу. За формою це дискретний часовий ряд**

$$F(t) = \{f_1(t), f_2(t), \dots, f_n(t)\}, \quad (2.4)$$

де $f_i(t)$ – потік даних з i -ї точки моніторингу в момент часу t .

2. **Агрегація та первинна фільтрація на рівні колектора. Реалізується оператором агрегації:**

$$A(F(t)) = \sum_{i=1}^n w_i \cdot f_i(t), \quad (2.5)$$

де w_i – вагові коефіцієнти для відповідних потоків даних.

3. **Аналіз даних для виявлення відхилень від норми.** Це робиться шляхом обчислення різниці між сукупними даними та базовим профілем мережевої активності $B(t)$:

$$D(t) = |A(F(t)) - B(t)|, \quad (2.6)$$

де $B(t)$ визначається на основі історичних даних.

4. **Використання правил Suricata.** Формалізується як функція прийняття рішення:

$$R(F(t)) = \begin{cases} 1, & \text{якщо виявлені аномалії,} \\ 0, & \text{в інакшому випадку.} \end{cases}, (2.7)$$

5. **Співвідношення виявлених аномалій із сформованими повідомленнями.** Описується відповідною функцією:

$$C(D(t), R(t)) = \varphi(D(t), R(t), H(t)), (2.8)$$

де $H(t)$ – історичні дані про попередні інциденти, а φ – функція кореляції, з урахуванням контексту.

Головною перевагою цього рішення є низький вплив на продуктивність мережі – загальний вплив на пропускну здатність не перевищує 1-2% від бази трафіку, що відповідає обмеженням промислових систем.

2.2 Поведінкова модель на основі NetFlow даних

2.2.1 Формалізація потоків даних

Дані NetFlow представлені як n -вимірний вектор ознак:

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, (2.9)$$

де компоненти вектора включають:

- $x_1(t)$ — кількість з'єднань за одиницю часу;
- $x_2(t)$ — кількість унікальних IP-адрес джерела;
- $x_3(t)$ — кількість унікальних IP-адрес призначення;
- $x_4(t)$ — об'єм трафіку (байт/с);
- $x_5(t)$ — кількість пакетів за одиницю часу;
- $x_6(t)$ — середній розмір пакету;
- $x_7(t), \dots, x_n(t)$ — інші релевантні метрики.

Кожну частину можна представити як часовий ряд із сезонною складовою:

$$x_i(t) = T_i(t) + S_i(t) + \varepsilon_i(t), (2.10)$$

де $T_i(t)$ — трендова складова, $S_i(t)$ — сезонна компонента з періодом P (зазвичай 24 години для промислових систем), а $\varepsilon_i(t)$ — випадкова компонента (шум).

2.2.2 Нормальна модель поведінки мережі

Для формалізації нормальної поведінки мережі використовується матриця переходу:

$$M = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}, \quad (2.11)$$

де p_{ij} — ймовірність переходу зі стану i в стан j , m — кількість дискретних станів системи. Стан системи визначається за допомогою кластеризації векторів ознак, наприклад, за допомогою алгоритму k-means:

$$S = \{s_1, s_2, \dots, s_m\}, \quad (2.12)$$

Кожен кластер характеризується своїм центроїдом:

$$c_i = \begin{bmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{in} \end{bmatrix}, \quad (2.13)$$

Базовий профіль нормальної активності $B(t)$ формується як часовий ряд очікуємих значень для кожного признаку:

$$B(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_n(t) \end{bmatrix}, \quad (2.14)$$

де для кожного i справедливо:

$$b_i(t) = \frac{1}{|T_t|} \sum_{t' \in T_t} x_i(t'), \quad (2.15)$$

при цьому T_t — множина історичних моментів часу, відповідаючих t .

2.2.3 Методи виявлення аномалій

Для виявлення аномалій використано Метод трьох сігма, де для кожного показника $x_i(t)$ формуються ковзні оцінки:

$$\mu_i(t) = \frac{1}{|W|} \sum_{j \in W} x_i(j) \quad (2.16), \quad \sigma_i(t) = \sqrt{\frac{1}{|W|} \sum_{j \in W} (x_i(j) - \mu_i(t))^2} \quad (2.17)$$

- де $\sigma_i(t)$ — ковзне стандартне відхилення для i -ї метрики на момент t ; вимірює розкид значень у вікні W .
- $\mu_i(t)$ — Ковзне середнє (moving mean) для i -ї метрики на момент t ; оцінює «базовий» рівень показника за останні $|W|$ спостережень
- $x_i(t)$ — поточне спостереження, яке **порівнюється** з порогами $\mu_i(t) \pm 3\sigma_i(t)$
- $x_i(j)$ — історичні значення ознаки у середині вікна; саме вони потрапляють у суму

Окрім класичного «трьох сігма» для локального швидкого скринінгу аномалій, також застосовано підхід на основі прогнозування часових рядів. Алгоритм працює так:

1. Декомпозиція
2. Побудова моделі залишків
3. Прогноз і довірчі інтервали
4. Критерій аномалій

Для декомпозиції використаємо формулу трьохкомпонентної декомпозиції 2,10. Після цього Залишки $\varepsilon_i(t)$ апроксимуються авторегресійною моделлю другого порядку (моделлю Юле Волкера 2 порядку):

$$\varepsilon_i(t) = \phi_{i,1}\varepsilon_i(t-1) + \phi_{i,2}\varepsilon_i(t-2) + \mu_i(t), \quad (2.18)$$

де $\phi_{i,1}$ і $\phi_{i,2}$ — оцінені коефіцієнти, а $\mu_i(t)$ — білий шум з дисперсією $\sigma_{u,i}^2$

На базі останніх 30 хв (з дискретизацією, наприклад, кожні 3 хв) прогнозуємо $K \leq 10$ майбутніх значень залишків $\hat{\varepsilon}_i(t+1), \dots, \hat{\varepsilon}_i(t+K)$.

Для кожного кроку прогнозу будемо $100(1-\alpha)\%$ довірчий інтервал:

$$\hat{x}_i(t+k) \pm z_{1-\frac{\alpha}{2}} \sigma_{u,i} \sqrt{1 + \sum_{m=1}^{k-1} \psi_m^2} \quad (2.19)$$

- де $\hat{x}_i(t+k) = T_i(t+k) + S_i(t+k) + \hat{\varepsilon}_i(t+K)$, (2.20)
- ψ_m — коефіцієнти імпульсної характеристики $AR(2)$

- $z_{1-\frac{\alpha}{2}}$ — квантиль нормального розподілу для $\alpha \in \{0.05, 0.01\}$.

Якщо фактичне спостереження $x_i(t+k)$ виходить за межі побудованого довірчого інтервалу ($\alpha=0,01 \div 0,05$),— відмічаємо аномалію.

На кожному edge-вузлі для кожної метрики підтримується короткочасне (ковзне) обчислення середнього значення та його відхилення за останнє вікно спостережень. Якщо поточне значення виходить за межі трьох стандартних відхилень від середнього, система одразу фіксує його як потенційну аномалію. Паралельно залишки від трендово-сезонної декомпозиції «чистого» сигналу моделюються авторегресійною моделлю другого порядку, на основі якої за останні півгодини прогнозуються кілька наступних значень із побудовою довірчих інтервалів. Окреме сплескове відхилення як за класичним «трьох сігма», так і за прогнозом поза інтервалом — також вважається аномальним. Завдяки низькій ентропії та чіткій циклічності промислових мереж такий подвійний підхід забезпечує високу точність виявлення при мінімумі хибних тривог.

Усі локальні «спрацьовування» негайно відправляються до центрального модуля координації, де вони зберігаються і корелюються з історією подібних інцидентів. Центральний алгоритм зважує статистичні тривоги та сигнатурні сповіщення Suricata, групує пов'язані події в єдині інциденти й ранжує їх за пріоритетом. Таким чином комбінуючи швидкість edge-детекції з глибиною централізованого аналізу, система дає аналітикам змогу оперативно відокремлювати реальні загрози від одноразових сплесків та ефективно реагувати на події в ICS-OT мережах.

2.3 Suricata IDS у складі системи моніторингу

2.3.1 Теоретичні основи правил Suricata

Suricata IDS використовує систему правил, засновану на сигнатурах і шаблонах мережевого трафіку. Кожне правило можна формалізувати як логічний предикат:

$$P(\text{packet}) = H(\text{header}) \wedge C(\text{content}) \wedge M(\text{metadata}), \quad (2.21)$$

где:

- $H(\text{header})$ описує властивості, пов'язані із заголовками пакету (IP, порти, протоколи);
- $C(\text{content})$ визначає умови, що відносяться до вмісту пакету;
- $M(\text{metadata})$ - додаткові метадані та контекст.

2.3.2 Спеціальні правила для промислових мереж

Для промислових мереж було розроблено спеціалізований набір правил Suricata відповідно до особливостей середовищ ОТ.

1. Правила виявлення аномального росту кількості з'єднань.

Правило визначено наступним чином:

```
alert tcp any any -> $INDUSTRIAL_ZONE any (msg:"Excessive connection attempts to industrial zone"; flow:to_server; threshold: type threshold, track by_src, count 10, seconds 60; classtype:attempted-recon; sid:3000001; rev:1;)
```

Математично, воно відповідає умові:

$$C_{\text{src}}(t, \Delta t) > \theta_{\text{conn}}, \quad (2.22)$$

де $C_{\text{src}}(t, \Delta t)$

— кількість з'єднань від точки походження src в інтервалі $[t, t + \Delta t]$, а θ_{conn} — порогов параметр.

2. Правила виявлення нетипових для ОТ-середовища протоколів.

Правило визначається наступним чином:

```
alert ip any any -> $INDUSTRIAL_ZONE any (msg:"Unusual protocol in industrial network"; flow:to_server; ip_proto:!1,!6,!17,!47,!50,!89,!132; classtype:protocol-command-decode; sid:3000002; rev:1;)
```

Правило базується на принципі позитивної безпеки. Формально:

$$P_{\text{allowed}}(\text{protocol}) = \begin{cases} 1, & \text{if } \text{protocol} \in \{\text{TCP}, \text{UDP}, \text{GRE}, \text{ESP}, \text{OSPF}, \text{SCTP}\}, \\ 0, & \text{в іншому випадку.} \end{cases}, \quad (2.23)$$

3. Правила для виявлення аномального ICMP-трафіку.

Формуються наступним чином:

alert icmp \$EXTERNAL_NET any -> \$INDUSTRIAL_ZONE any (msg:"ICMP Ping to Industrial Zone"; itype:8; threshold: type threshold, track by_src, count 5, seconds 60; classtype:attempted-recon; sid:3000003; rev:1;)

Правило фіксує аномалії, якщо

$$I_{src}(t, \Delta t) > \theta_{icmp}, (2.24)$$

де $I_{src}(t, \Delta t)$ — кількість ICMP-пакетів від джерела src за інтервал $[t, t + \Delta t]$.

4. Правила для виявлення комунікацій між сегментами, які не повинні взаємодіяти.

Правило формулюється наступним чином:

alert ip \$LEVEL3_NET any -> \$LEVEL1_NET any (msg:"Illegal communication between Level 3 and Level 1"; flow:to_server; classtype:policy-violation; sid:3000004; rev:1;)

Математично визначається як:

$$C_{allowed}(src, dst) = \begin{cases} 1, & \text{якщо } (L(src), L(dst)) \in AllowedPairs, \\ 0, & \text{в іншому випадку.} \end{cases} (2.25)$$

де функція $L(x)$ повертає рівень Purdue для мережевого сегменту x.

5. Правила для виявлення ТСП-сканування.

Правило визначається наступним чином:

alert tcp any any -> \$INDUSTRIAL_ZONE any (msg:"TCP Port Scan against Industrial Zone"; flags:S; threshold: type threshold, track by_src, count 5, seconds 10; flow:to_server,no_stream; classtype:attempted-recon; sid:3000005; rev:1;)

Формально, умова подібного сканування записується як:

$$S_{src}(t, \Delta t) > \theta_{scan}, (2.26)$$

де $S_{src}(t, \Delta t)$ — кількість TCP SYN-пакетів от джерела src за інтервал $[t, t + \Delta t]$.

2.4 Алгоритми кореляції та агрегації подій

2.4.1 Теоретична модель кореляції подій

Щоб підвищити точність виявлення складних аномалій, ми використовуємо систему кореляції подій і формалізуємо її за допомогою теорії графів. Події представлені як вершини графа:

$$G_{\text{events}} = (V_{\text{events}}, E_{\text{relations}}), (2.27)$$

де V_{events} — множина подій, а $E_{\text{relations}}$ — множина зв'язків між подіями, що відображають їхню каузальну залежність.

Кожна подія характеризується вектором атрибутів:

$$e_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix}, (2.28)$$

Сила зв'язку між подіями обчислюється як функція від їхньої семантичної близькості та часової кореляції:

$$w_{ij} = \alpha \cdot S(e_i, e_j) + (1 - \alpha) \cdot T(t_i, t_j), (2.29)$$

де:

- $S(e_i, e_j)$ — семантична подібність подій,
- $T(t_i, t_j)$ — часова кореляція, що залежить від різниці у часі між подіями,
- α — параметр, що визначає відносну важливість семантичної та часової складових.

2.4.2 Алгоритм кластеризації подій

Для групування пов'язаних подій використовується алгоритм ієрархічної кластеризації:

1. Початково кожна подія утворює окремий кластер:

$$C_i = \{e_i\}, (2.30)$$

2. На кожній ітерації об'єднуються два найближчі кластери, де відстань між кластерами визначається за формулою:

$$d(C_i, C_j) = \min_{e_a \in C_i, e_b \in C_j} d(e_a, e_b), (2.31)$$

при цьому відстань між подіями задається як:

$$d(e_a, e_b) = 1 - w_{ab}, (2.32)$$

3. Цей процес триватиме, доки не буде досягнуто порогове значення відстані або задана кількість кластерів.

Для виявлення причинно-наслідкових зв'язків використовується Байєсівська мережа, яка описується наступним співвідношенням:

$$P(E) = \prod_{i=1}^n P(e_i | pa(e_i)), (2.33)$$

де $pa(e_i)$ — множина батьківських вершин (подій-передумов) для події e_i .

2.4.3 Агрегація та ранжування інцидентів

Для агрегації виявлених аномалій у інциденти вищого рівня використовується функція агрегації:

$$I = A(E, R) = \{(e_i, r_j) \mid e_i \in E, r_j \in R, C(e_i, r_j) > \theta_c\}, (2.34)$$

де:

- E — множина виявлених аномалій,
- R — множина правил Suricata,
- $C(e_i, r_j)$ — функція кореляції між аномалією та правилом,
- θ_c — поріг кореляції.

Ранжування інцидентів виконується за наступною формулою:

$$\text{Score}(I) = \sum_{(e_i, r_j) \in I} w_i \cdot \text{Severity}(r_j) \cdot \text{Confidence}(e_i), (2.35)$$

де:

- w_i — вага аномалії,
- $\text{Severity}(r_j)$ — рівень критичності правила,
- $\text{Confidence}(e_i)$ — рівень впевненості у виявленні аномалії.

2.5 Обґрунтування вибору одновимірного детектування аномалій

На відміну від корпоративних ІТ-мереж, де користувацька поведінка може бути дуже різноманітною, промислові системи характеризуються низькою ентропією та статичністю. Ключовими факторами, що обґрунтовують вибір одновимірного детектування аномалій, є:

2.5.1 Математичне обґрунтування статичності конфігурації та трафіку

Промислові системи функціонують згідно з заздалегідь визначеними алгоритмами, з чітко регламентованими шаблонами комунікацій. Як зазначається в NIST SP 800-82, до 95 % мережевих взаємодій в ОТ-середовищах є циклічними та

передбачуваними, з фіксованими інтервалами між взаємодіями. Математично це може бути описано через низьку ентропію мережових взаємодій:

$$H(X) = - \sum_{i=1}^n p(x_i) \log p(x_i), \quad (2.36)$$

де X — множина можливих мережових взаємодій, а $p(x_i)$ — ймовірність виникнення взаємодії x_i . Для ОТ-систем характерним є:

$$H_{ICS}(X) \ll H_{IT}(X), \quad (2.37)$$

2.5.2 Формалізація автоматизованої природи комунікацій

Відсутність "живих користувачів" дозволяє моделювати комунікації як процеси Маркова з низькою ентропією переходів:

$$P(X_{t+1} = j \mid X_t = i) = p_{ij}, \quad (2.38)$$

з властивістю, що для кожного i існує таке j , що $p_{ij} \approx 1$.

За даними ICS-CERT понад 90 % легітимних мережових взаємодій в ОТ-системах ініціюються автоматично за розкладом або на основі тригерів, що відображено у низькій умовній ентропії:

$$H(X_{t+1} \mid X_t) = - \sum_{i,j} p(i,j) \log \frac{p(i,j)}{p(i)}, \quad (2.39)$$

2.5.3 Теоретичне обґрунтування критичності відхилень

В умовах промислових систем практично будь-яке відхилення від нормальної поведінки може свідчити про кіберінцидент або технічну несправність, що формалізується наступною апріорною ймовірністю:

$$P(\text{Anomaly} \mid \text{Deviation}) \approx 1, \quad (2.40)$$

За даними Dragos Research, у 78 % підтверджених інцидентів в ICS-мережах першим індикатором компрометації було виявлення аномалій мережевого трафіку за межами стандартних шаблонів. З точки зору теорії прийняття рішень оптимальною стратегією є негайне сповіщення при виявленні відхилень, якщо співвідношення вартості пропуску інциденту (C_{miss}) до вартості хибного спрацьовування (C_{false}) задовольняє нерівність:

$$\frac{C_{\text{miss}}}{C_{\text{false}}} > \frac{P(\text{Normal} | \text{Deviation})}{P(\text{Anomaly} | \text{Deviation})}, (2.41)$$

Таким чином, замість використання багаторівневого скорінгу, що традиційно застосовується у системах виявлення вторгнень, доцільним є використання бінарної класифікації «норма–аномалія» з безпосереднім сповіщенням при виявленні відхилень.

2.6 Метрики ефективності та оптимізація сигнатурних правил

2.6.1 Формальне визначення метрик ефективності

Для оцінки ефективності запропонованого підходу використовуються наступні метрики:

1. **Точність (Precision)** – частка правильно ідентифікованих аномалій серед усіх виявлених:

$$\text{Precision} = \frac{TP}{TP + FP}, (2.42)$$

2. **Повнота (Recall)** – частка виявлених аномалій серед усіх реальних аномалій:

$$\text{Recall} = \frac{TP}{TP + FN}, (2.43)$$

3. **F1-міра** – гармонійне середнє точності та повноти:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, (2.44)$$

4. **Час виявлення (MTTD)** – середній час від початку аномалії до її виявлення:

$$\text{MTTD} = \frac{1}{n} \sum_{i=1}^n (t_{\text{detect},i} - t_{\text{start},i}), (2.44)$$

5. **ROC-крива** – залежність між True Positive Rate та False Positive Rate при різних порогових значеннях, де:

$$TPR = \frac{TP}{TP + FN}, (2.45) \quad FPR = \frac{FP}{FP + TN}, (2.46)$$

6. **AUC (Area Under Curve)** – площа під ROC-кривою, що характеризує загальну ефективність моделі незалежно від порогового значення.

2.6.2 Методи оптимізації параметрів

Для оптимізації параметрів системи використовується функція втрат:

$$L(\theta) = \alpha \cdot FPR(\theta) + (1 - \alpha) \cdot [1 - TPR(\theta)],, (2.47)$$

де θ – вектор параметрів системи, α – коефіцієнт, що визначає відносну важливість мінімізації хибно-позитивних та хибно-негативних спрацьовувань.

Оптимальні параметри знаходяться шляхом мінімізації функції втрат:

$$\theta^* = \arg \min_{\theta} L(\theta), (2.48)$$

Для багатокритеріальної оптимізації застосовується парето-оптимальний підхід, який дозволяє знайти набір параметрів, що не можуть бути покращені за одним критерієм без погіршення за іншим.

2.6.3 Валідація ефективності на тестових сценаріях

Для валідації ефективності запропонованого підходу розроблено набір тестових сценаріїв, що моделюють різні типи аномалій:

1. **Сканування мережі та portscans** — моделюється як різке зростання кількості підключень з одного джерела до багатьох цільових портів.
2. **Аномальний міжсегментний трафік** — моделюється як поява підключень між сегментами, які зазвичай не взаємодіють.
3. **Тунелювання через легітимні протоколи** — моделюється як зміна статистичних характеристик легітимного трафіку (ентропія, розподіл розмірів пакетів).
4. **Компрометація системи управління** — моделюється як поява нових типів команд або зміна частоти існуючих.

Для кожного сценарію проводиться оцінка ефективності за визначеними метриками та порівняння з базовими методами.

2.6.4 Взаємозв'язок між одновимірним детектуванням та одноранговою архітектурою

Вибір одновимірного детектування аномалій органічно доповнює концепцію однорангової архітектури системи. Цей взаємозв'язок обґрунтовується наступними факторами:

1. Спрощення автономного прийняття рішень - одновимірний підхід дозволяє кожному вузлу в одноранговій мережі приймати рішення на основі чітких порогових значень без необхідності складної багатовимірної аналітики.
2. Зниження обчислювальної складності - за рахунок одновимірного аналізу кожен вузол може ефективно функціонувати без потреби в централізованих обчислювальних ресурсах, що повністю відповідає принципу децентралізації в одноранговій архітектурі.
3. Підвищення швидкодії виявлення - поєднання одновимірного детектування з одноранговою архітектурою забезпечує мінімальну затримку між виникненням аномалії та її виявленням (MTTD), що можна формалізувати як:

$$MTTD_{\text{однорангова}} < MTTD_{\text{ієрархічна}}$$
4. Статистична обґрунтованість - для промислових систем з їх високою ймовірностями переходу, однорангова архітектура з одновимірним детектуванням забезпечує оптимальне співвідношення між простотою реалізації та ефективністю виявлення, що підтверджується показником:

$$P(\text{Anomaly}|\text{Deviation}) \approx 1$$

2.7 Очікувані результати та обмеження методики

2.7.1 Прогнозована ефективність для різних типів аномалій

На основі теоретичного аналізу та попередніх експериментів очікуються наступні показники ефективності запропонованого підходу:

Таблиця 2.2 – Прогнозована ефективність для різних типів аномалій

Тип аномалії	Очікувана точність (%)	Очікувана повнота (%)	MTTD (хв)
Сканування мережі	>95	>90	<3
Аномальний міжсегментний трафік	>90	>85	<5
Витік даних	>85	>80	<10
Тунелювання через легітимні протоколи	>80	>75	<15
Компрометація системи управління	>85	>80	<8

2.7.2 Теоретичні обмеження підходу

Запропонована модель має ряд теоретичних обмежень:

1. Обмеження однорангової архітектури - хоча однорангова система забезпечує підвищену надійність і швидкодію, вона може зіткнутися з проблемами синхронізації та узгодженості даних між вузлами, що формалізується як:

$$\Delta t_{sync} \propto N_{nodes},, (2.49)$$

де Δt_{sync} - час, необхідний для синхронізації даних, а N_{nodes} - кількість вузлів.

2. Обмеження одновимірного детектування – хоча підхід є ефективним для статичних ОТ-середовищ, він може бути менш точним для виявлення складних багатовимірних аномалій.
3. Залежність від якості базового профілю – ефективність виявлення значно залежить від точності формування нормального стану системи, що може бути формалізовано як:

$$\text{Error}(\text{Detection}) \propto \text{Error}(\text{Baseline}), (2.50)$$

4. Обчислювальна складність – для великих мереж алгоритми кореляції мають складність $O(n^2)$, де n — кількість подій, що може бути обмеженням для обробки в режимі реального часу.

2.7.3 Перспективи подальшого розвитку

Ґрунтуючись на отриманих результатах дослідження та враховуючи обмеження виявлені під час тестування системи, можна визначити такі перспективні напрямки розвитку методики:

Інтеграція методів глибокого навчання для виявлення більш складних патернів аномалій

У поточній реалізації системи використовуються переважно традиційні методи машинного навчання для виявлення аномалій, однак, як показує сучасна практика, глибоке навчання демонструє значно вищу ефективність при роботі зі складними патернами в промислових даних. Зокрема, перспективними є такі напрямки:

- **Рекурентні нейронні мережі (LSTM)** — дослідження підтверджують їх ефективність (до 94% точності) для виявлення аномалій у послідовностях команд і часових рядах промислових систем SCADA. На відміну від використаних у роботі методів, LSTM здатні враховувати довгострокові залежності в даних.
- **Автоенкодери** демонструють високу ефективність у виявленні тонких відхилень та поступового дрейфу параметрів системи з часом. Глибокі автоенкодери на основі RBM (restricted Boltzmann machines) показали високу ефективність для раннього виявлення аномалій в промислових системах, наприклад, вітрових турбінах.
- **Гібридні підходи**, що поєднують ізоляційні ліси (Isolation Forest) для швидкого виявлення аномалій в реальному часі з автоенкодерами для глибинного аналізу та валідації, демонструють найвищу точність виявлення (до 85-95% для різних типів атак) при збереженні обчислювальної ефективності.

Розробка адаптивних моделей базового профілю

Як було висвітлено в розділі 2.4, створення базового профілю є ключовим етапом для ефективного виявлення аномалій. Подальшими напрямками розвитку є:

- **Автоматичне оновлення базового профілю** з урахуванням легітимних змін у конфігурації системи. Згідно з дослідженнями, процес створення базового профілю має включати: збір даних, аналіз і визначення нормальної поведінки, встановлення базової лінії, постійний моніторинг та регулярне оновлення.
- **Диференційований підхід до базових профілів** для різних типів пристроїв. Дослідження показують, що необхідно розробляти окремі базові профілі для мережевого обладнання, кінцевих пристроїв, систем користувачів та специфічних промислових протоколів (Modbus, Profinet, EtherCAT).
- **Інтеграція з механізмами безпеки Microsoft Defender** для автоматичного створення та оновлення профілів безпеки на основі галузевих стандартів та

конкретних вимог організації, що дозволить досягти кращої сумісності з існуючими системами.

Удосконалення алгоритмів кореляції

Дослідження показало, що кореляційний аналіз є потужним інструментом для виявлення аномалій у промислових системах, проте існуючі алгоритми мають високу обчислювальну складність. Перспективними напрямками є:

- **Впровадження методів латентного кореляційного аналізу (LCM)**, які трансформують кореляційні матриці у вектори латентних кореляцій, що суттєво знижує обчислювальну складність з $O(n^2)$ до $O(n \log n)$ при збереженні точності виявлення аномалій до 95%.
- **Застосування техніки рухомого вікна (sliding window)** з динамічним налаштуванням розміру вікна для аналізу часових рядів, що дозволяє оптимізувати обчислювальні ресурси та підвищити ефективність виявлення аномалій у промислових системах.
- **Імплементація методології VLSTM (Variational Long Short-Term Memory)** для зменшення розмірності даних та ефективного виділення прихованих кореляцій між пристроями, що особливо важливо для систем з великою кількістю сенсорів та виконавчих механізмів.

Розширення системи для підтримки гетерогенних промислових середовищ

Сучасні промислові середовища характеризуються різноманітністю обладнання та протоколів, що ускладнює впровадження єдиних систем моніторингу. Перспективними напрямками розвитку є:

- **Підтримка ширшого спектру промислових протоколів**, включаючи традиційні послідовні протоколи (Modbus, Profibus), промислові Ethernet-протоколи (EtherNet/IP, Profinet) та бездротові протоколи (WirelessHART).
- **Розробка єдиного шлюзу для різних типів даних з різних середовищ (OT/IT/IoT)**, що дозволить централізовано збирати та аналізувати дані незалежно від їх джерела та формату.
- **Впровадження адаптивних методів виявлення аномалій**, які враховують специфіку протоколів та обладнання. Дослідження показують, що

ефективність виявлення аномалій значно зростає при врахуванні особливостей конкретних промислових середовищ.

- **Розробка масштабованих рішень** для моніторингу та аналізу безпеки великомасштабних промислових мереж з можливістю обробки різнорідних даних (статичних та часових рядів, дискретних та неперервних).

Ці напрямки розвитку дозволять підвищити ефективність методики виявлення аномалій у промислових мережах, знизити кількість хибнопозитивних спрацьовувань та забезпечити більш надійний захист критичної інфраструктури від кіберзагроз.

3 Експериментальне дослідження

3.1 Побудова лабораторного стенда

У якості експериментальної платформи було використано віртуалізоване середовище на базі серверу з параметрами:

- **CPU:** AMD Ryzen 9 5950X (16 cores / 32 threads)
- **RAM:** 128 GiB DDR4-3200
- **Диски:** NVMe SSD 4тб

Мережевий сегмент 192.168.100.0/24 і 192.168.10.0/24 розбитий за моделлю Purdue на рівні 2–3 з VLAN-тегуванням і mirror-портом для передавання копії трафіку на аналізуючий вузол.

Таблиця 3.1 – Архітектура середовища

IP-адреса	Роль вузла	Протокол	Завдання
192.168.10.10	Suricata IDS	AF_PACKET, Mirror-інтерфейс	Обробка трафіку в мережі
192.168.10.20	Master (SCADA-сервер)	Modbus TCP, DNP3, EtherNet/IP	Ініціює опитування slave-пристроїв, візуалізує дані, зберігає історію, обробляє події та тривоги
192.168.100.40	DNP3 Outstation (Substation)	DNP3	Моніторинг електричної підстанції: стан вимикачів, захистів, трансформаторів
192.168.100.50	Modbus TCP Slave (Production)	Modbus TCP	Контроль конвеєра виробничої лінії, відстеження датчиків присутності продукції
192.168.100.60	Modbus TCP Slave (Pumping)	Modbus TCP	Управління насосною станцією: тиск, витрата, рівні в резервуарах
192.168.100.70	EtherNet/IP Slave (Robotic)	EtherNet/IP	Керування роботизованою системою: демонструє гетерогенність SCADA-середовища

3.1.1 Апаратно-віртуальна платформа

Усі віртуальні машини розгорнуті на Esxi VE, встановленому на головному сервері.

Кожній VM виділено по **2 vCPU, 2 GiB RAM і 20 GiB** диску. Нижче наведено реальні дані по споживанню ресурсів трьох ключових VM (за результатами neofetch):

Таблиця 3.2 – Ресурси VM

Вузол	Виділено ресурсів	CPU-посадка (vCPU)
PLC (192.168.100.30)	2 vCPU, 2 GiB RAM, 20 GiB диск	2
Suricata (192.168.100.20)	2 vCPU, 2 GiB RAM, 20 GiB диск	2
Slave (30, 40, 50, 60, 70)	2 vCPU, 2 GiB RAM, 20 GiB диск	2

За спостереженнями, навіть під навантаженням атаками пік використання RAM не перевищує 60 % від виділеного, а процесорні ядра залишаються вільними для розгортання додаткових сервісів. Це підтверджує вибір скромних розмірів VM для емітації ОТ-завантаження без перевитрати ресурсів хост-машини. Для автоматичного розгортання сервісів на slave-вузлах використовується скрипт `deploy_all_slaves.sh`.

3.1.2 Логічна топологія ОТ-сегмента

Мережа поділена на дві зони (Purdue L2–L3), master і slave взаємодіють лише через відповідні порти (502, 20000, 44818). На рівні 2 відбувається збір проміжних даних, на рівні 3 — централізована обробка та збереження.

Всі міжвузлові з'єднання проходять через bridge із mirror-функціоналом; доступ до management-портів (22, 8080) обмежений правилами OPNsense.

3.2 Генерація еталонного й атакуючого трафіку

Для кожного slave-вузла підготовлено два “генератори” трафіку — **normal** для еталонного поведінкового профілю та **attack** для перевірки правил Suricata. Кожен скрипт налаштований під роль ПЛК/RTU, які він емулює, і містить унікальний набір змодельованих атак під конкретний протокол.

192.168.100.30 – HVAC (Modbus TCP)

- **Normal:**

Скрипт `normal_traffic_30.sh` циклічно генерує показники температури, вологості, повітряного потоку та CO₂ по чотирьох зонах і записує їх у реєстри Master через Modbus TCP. Пауза між циклами 15–25 с. Логи — `/var/log/normal_traffic_hvac.log`.

- **Attack:**

`attack_traffic_30.sh` випадково обирає атаку з набору:

- **1000001** – неавторизований Modbus (TCP-з'єднання на 502 порт)
- **1000002** – порт-скан ≥ 10 SYN/60 с
- **1000006** – ICMP-флуд (> 5 пакетів/60 с)
- **1000012** – TCP-флуд (> 20 з'єднань/10 с)
- **1000007–9** – записи FC05/FC06/FC16

Усі кроки логуються у `/var/log/attack_traffic_hvac.log`.

- **Systemd-unit:**

`hvac-traffic.service.txt` автоматично стартує `normal_traffic_30.sh` після завантаження мережі.

192.168.100.40 – Substation (DNP3)

- **Normal:**

`normal_traffic_40.sh` імітує опитування стану вимикачів, захистів і трансформаторів через DNP3-outstation. Інтервал — 10–20 с, лог — `/var/log/normal_traffic_substation.log`.

- **Attack:**

`attack_traffic_40.sh` випадково виконує:

- **1000014** – неавторизований DNP3 WRITE (функція 1)
- **1000015** – TFTP WRQ на порт 69
- **1000002** – порт-скан
- **1000003** – спроба SSH/HTTP/HTTPS у ОТ

Усі атаки фіксуються у `/var/log/attack_traffic_substation.log`.

- **Systemd-unit:**

substation-traffic.service.txt запускає normal_traffic_40.sh після network-online.target.

192.168.100.50 – Production Line (Modbus TCP)

- **Normal:**

normal_traffic_50.sh передає статуси конвеєра, роботів, пакувальної лінії (coils), швидкість виробництва, кількість деталей, показники якості, температури двигунів і споживану потужність (holding registers)

- **Attack:**

attack_traffic_50.sh містить:

- **1000001** – неавторизований Modbus до інших slave
- **1000013** – lateral movement (SMB/HTTP/HTTPS/445 між OT)
- **1000007–9** – Modbus WRITE FC05/06/16
- **1000012** – TCP-flood
- **1000007–9** – саботаж (“production_sabotage”): запис небезпечних швидкостей і вимкнення захистів

Всі події — у /var/log/attack_traffic_production.log.

- **Systemd-unit:**

production-traffic.service.txt контролює запуск normal_traffic_50.sh.

192.168.100.60 – Pumping Station (Modbus TCP)

- **Normal:**

normal_traffic_60.sh передає статус насоса, тиск на вході/виході, витрату, рівні в резервуарах, температури й вібрації насосів. Інтервал 12–20 с, лог — /var/log/normal_traffic_pumping.log.

- **Attack:**

attack_traffic_60.sh підтримує сценарії:

- **1000001** – неавторизований Modbus
- **1000013** – lateral movement SMB/HTTP
- **1000010–11** – Modbus broadcast (Unit ID=0,255)
- **1000012** – TCP-flood

- **1000007–9** – саботаж: встановлення критичного тиску, вимкнення захистів, максимальна швидкість
- **1000016** – SMB-write (FF 53 4D 42 05)
Логи — /var/log/attack_traffic_pumping.log.

- **Systemd-unit:**

pumping-traffic.service.txt забезпечує старт нормального скрипта.

192.168.100.70 – Robotic System (EtherNet/IP)

- **Normal:**

normal_traffic_70.sh емулює EtherNet/IP-обмін: позиції 4 роботів, статуси, швидкості, поточні завдання (custom CIP message).

- **Attack:**

attack_traffic_70.sh реалізує:

- **1000099** – неавторизований EtherNet/IP (TCP 44818)
- **1000002** – порт-скан
- **1000003** – management access (22,80,443,8080)
- **1000012** – TCP-flood
- **1000018** – CIP Write Tag (0x4B) пакет
- **1000018** – саботаж роботів: критичні координати (9999,9999,9999), максимальна швидкість

Журнали — /var/log/attack_traffic_robot.log.

- **Systemd-unit:**

robot-traffic.service.txt автоматизує старт нормального трафіку.

3.3 Налаштування Suricata IDS

У стенді використовується **Suricata 7.0.3** на Ubuntu 24.04 для виявлення аномалій у ОТ-мережі. Основні налаштування:

- **Режим AF_PACKET**

Suricata читає трафік з mirror-інтерфейсу ens36 через AF_PACKET:

- af-packet:
- - interface: ens36
- cluster-id: 99

- cluster-type: cluster_flow
- defrag: yes
- use-mmap: yes
- mmap-locked: yes
- tpacket-v3: yes

Це забезпечує мінімальні пропуски навіть під високим навантаженням.

- **EVE-журнал**

Увімкнено JSON-логування у /var/log/suricata/eve.json:

- outputs:
 - - eve-log:
 - enabled: yes
 - filetype: regular
 - filename: eve.json
 - types:
 - - alert
 - - http
 - - dns
 - - tls
 - - stats

Такі логи легко індексуються Filebeat → Elasticsearch → Grafana.

- **Інтеграція з Elastic / Grafana**

Через Filebeat подаємо eve.json в індекс suricata-*. У Grafana налаштовані дашборди:

- **Alerts Over Time** за SID, severity
- **Protocol Breakdown** (Modbus, DNP3, CIP)
- **Top Talkers** та heatmap порушень за сегментами

3.3.1 Структура власного rule-set

Набір правил охоплює всі ключові сценарії поведінки та атак у SCADA-мережі та складається з 18 правил, згрупованих за п'ятьма категоріями:

Таблиця 3.3 – Опис категорій правил Suricata

Категорія	Назва	Опис категорії
0	Базовий контроль Modbus-TCP	Забезпечує, що Modbus-трафік походить лише від легітимного master-хоста і тільки до slave-пристроїв; блокує несанкціоновані Modbus-запити, порт-сканування та спроби доступу до портів управління.
1	Контури сегментації / периметр	Слідкує за міжзонним трафіком ОТ: фіксує будь-який не-ICMP трафік, що покидає або входить в ОТ-сегмент, а також надмірне використання ICMP (ping-sweep).
2	Специфіка Modbus (записи/широковещалки)	Виявляє несанкціоновані записи у Modbus-ПЛК (FC05, FC06, FC16) та широковещальні запити Unit ID=0 і 255, які можуть порушити роботу всіх slave-пристроїв.
3	Мережеві аномалії / L2	Фіксує нетипові L2-шаблони в ОТ: надто часті TCP-з'єднання і небажані SMB/HTTP/HTTPS-сесії між вузлами всередині ОТ-зони.
4	DNP3 / TFTP / SMB / EtherNet/IP	Спеціалізовані правила для DNP3-WRITE, TFTP WRQ, SMB-write, ARP-флуд/скан, а також для запису тегів у EtherNet/IP (Explicit Write Tag) та загального CIP-трафіку.

Нижче більш детально розглянута кожна категорія і описана потрібність відповідних правил.

Категорія 0: Базовий контроль Modbus-TCP

Забезпечує, що Modbus-трафік походить лише від легітимного master-хоста і тільки до slave-пристроїв.

1. **SID 1000001 – Неавторизований Modbus-клієнт**
2. alert tcp !\$MASTER_IP any -> !\$MASTER_IP 502 (...)

Навіщо: блокує будь-які SYN/ACK/PUSH на порт 502 від хостів, які не є master (192.168.100.20). Це перший бар'єр проти сторонніх клієнтів, які намагаються напругу опитати slave-ПЛК.

3. **SID 1000002 – Port-scan**

4. alert tcp any any -> any any (flags:S; threshold: track by_src, count 10, seconds 60; ...)

Навіщо: виявляє сканування портів (≥ 10 SYN за хвилину), забороняючи розвідувальні спроби знайти промислові сервіси.

5. **SID 1000003 – Використання management-портів**

6. alert tcp any any -> \$HOME_NET [22,80,443,8080] (...)

Навіщо: фіксує підключення до SSH/HTTP/HTTPS ззовні в ОТ-зону. Забороняє несанкціонований доступ до інтерфейсів адміністрування.

Категорія 1: Контури сегментації / периметр

Стежить за міжзонним трафіком і допомагає підтримувати строгий периметр ОТ-сегмента.

4. **SID 1000004 – Трафік, що покидає ОТ (крім ICMP)**

5. alert ip \$HOME_NET any -> \$EXTERNAL_NET any (ip_proto:!1; ...)

Навіщо: фіксує будь-який не-ICMP трафік, що йде з ОТ у зовнішню мережу.

6. **SID 1000005 – Трафік, що входить в ОТ (крім ICMP)**

7. alert ip \$EXTERNAL_NET any -> \$HOME_NET any (ip_proto:!1; ...)

Навіщо: аналогічно, фіксує не-ICMP з'єднання ззовні до ОТ-зони — важливо для виявлення ін'єкції пакетів.

8. **SID 1000006 – Надмірний ICMP (ping-sweep)**

9. alert icmp any any -> any any (threshold: track by_src, count 5, seconds 60; ...)

Навіщо: блокує інтенсивні пінг-сканування, які можуть бути попередньою фазою атаки.

Категорія 2: Специфіка Modbus (записи/широковещалки)

Стежить за небезпечними командами запису та широковещальними запитам.

7. **SID 1000007 – Запис регістрів FC05 (Coil Write)**

8. alert modbus !\$MASTER_IP any -> \$SLAVE_IPS 502 (modbus:function 5; ...)

Навіщо: забороняє стороннім вузлам змінювати дискретні виходи PLC.

9. **SID 1000008 – Запис регістрів FC06 (Single Register Write)**

10. alert modbus !\$MASTER_IP any -> \$SLAVE_IPS 502 (modbus:function 6; ...)

Навіщо: аналогічно, перехоплює несанкціоновані зміни одиночних регістрів.

11.SID 1000009 – Запис регістрів FC16 (Multiple Registers Write)

12.alert modbus !\$MASTER_IP any -> \$SLAVE_IPS 502 (modbus:function 16; ...)

Навіщо: виявляє масові зміни встановлень у кількох регістрах за один запит.

13.SID 1000010 – Широковещальний Unit ID = 0

14.alert modbus any any -> \$SLAVE_IPS 502 (modbus:unit 0; ...)

Навіщо: broadcast-запити Unit ID=0 можуть порушити роботу всіх пристроїв.

15.SID 1000011 – Широковещальний Unit ID = 255

16.alert modbus any any -> \$SLAVE_IPS 502 (modbus:unit 255; ...)

Навіщо: аналогічно, захищає від broadcast на Unit ID=255, небезпечного для мережі.

Категорія 3: Мережеві аномалії / L2

Виявляє нетипові мережеві шаблони всередині ОТ-зони.

12.SID 1000012 – Надто часті з'єднання (≥ 20 pkt/10 с)

13.alert tcp any any -> \$HOME_NET any (detection_filter: track by_src, count 20, seconds 10; ...)

Навіщо: фіксує TCP-флуд або DDoS-фази всередині ОТ.

14.SID 1000013 – SMB/HTTP/HTTPS між ОТ-вузлами

15.alert tcp \$HOME_NET any -> \$HOME_NET [80,443,445] (...)

Навіщо: забороняє небезпечну взаємодію SMB/HTTP між PLC/RTU, яка не належить до контрольної архітектури.

Категорія 4: DNP3 / TFTP / SMB / EtherNet/IP

Обробляє специфічні протоколи та L2-атаки.

14.SID 1000014 – Неавторизований DNP3 WRITE (func=1)

15.alert tcp !\$MASTER_IP any -> \$SLAVE_IPS 20000 (dnp3_func:1; ...)

Навіщо: захищає підстанції від сторонніх команд керування.

16.SID 1000015 – TFTP WRQ (opcode 02)

17.alert udp \$HOME_NET any -> \$SLAVE_IPS 69 (content:"|00 02|"; ...)

Навіщо: виявляє спроби завантажити прошивку у RTU через TFTP.

18.SID 1000016 – SMB-write всередині ОТ

19.alert tcp \$HOME_NET any -> \$HOME_NET 445 (content:"|FF 53 4D 42 05|"; ...)

Навіщо: блокує несанкціоноване копіювання/запис файлів через SMB у OT.

20.SID 1000017 – ARP-флуд / скан

21.alert ip any any -> any any (content:"|00 01|"; offset:20; depth:2; detection_filter: track by_src, count 30, seconds 10; ...)

Навіщо: фіксує агресивний ARP-трафік, який може призвести до ARP-спуфінгу.

22.SID 1000018 – EtherNet/IP Explicit Write Tag (service 0x4B)

23.alert tcp any any -> \$HOME_NET 44818 (content:"|4B|"; offset:40; depth:10; ...)

Навіщо: перехоплює записи тегів у CIP, які можуть змінити поведінку роботизованої системи.

24.SID 1000099 – EtherNet/IP Traffic

25.alert tcp any any -> \$HOME_NET 44818 (...)

Навіщо: загальне відстеження EtherNet/IP-сесій для подальшого аналізу.

Кожне правило націлене на специфічні загрози для розглянутої архітектури master-slave, забезпечуючи багаторівневий захист: від базового контролю Modbus до глибокої інспекції DNP3, EtherNet/IP і L2-аномалій.

```
root@suricata ~# suricata -T -c /etc/suricata/suricata.yaml -S /etc/suricata/rules/ot.rules
i: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
i: suricata: Configuration provided was successfully loaded. Exiting.
root@suricata ~#
```

Рисунок 3.1 – Перевірка правильності правил Suricata і конфігурації

3.4 Результати та аналіз

3.4.1 Якість детектування

У всіх протестованих сценаріях Suricata 7.0.3 досягла близько **100 %**

Precision та **100 % Recall** завдяки повному покриттю rule-set(іноді були False Positive реакції, які були із-за не зовсім точності правил детектуванні і простоти скриптів для генерації трафіку). Час реакції (MTTD) становить **0.5 – 1 с** для більшості правил, оскільки Suricata генерує alert одразу при отриманні пакета, що відповідає умові правила.

Особливий випадок — правило **1000012** (“Abnormal connection rate”), яке спрацьовує після накопичення ≥ 20 пакетів протягом 10 с; тому фактичний MTTD = **10 с (пороговий інтервал) + 0.5 – 1 с** на обробку та логування.

Таблиця 3.4. – Оцінка якості детектування

Тип аномалії	Precision	Recall	MTTD
Сканування портів (SID 1000002)	~100 %	100 %	0.5 – 1 с
Неавторизовані Modbus-записи (SID 1000007–9)	100 %	100 %	0.5 – 1 с
Спроби доступу до management-портів (1000003)	~100 %	100 %	0.5 – 1 с
DNP3 WRITE не-MASTER-ом (1000014)	100 %	100 %	0.5 – 1 с
TFTP WRQ (1000015)	100 %	100 %	0.5 – 1 с
EtherNet/IP Write Tag (1000018)	100 %	100 %	0.5 – 1 с
Аномальна частота з’єднань (1000012)	100 %	100 %	10 с + 0.5 – 1 с

3.4.2 Навантаження на ресурси

Під час синтетичних атак та пікового еталонного трафіку Suricata споживала ресурси близько:

- **ЦПУ:** 55–70 % від двох vCPU
- **Оперативна пам’ять:** 800–1 200 MiB (разом із Filebeat та додатковими демонами)
- **EVE-JSON логи:** ≈ 50 MiB/год $\rightarrow \approx 1.2$ GiB/доба

Завдяки режиму AF_PACKET з tpacket-v3 і mmap усі пакети обробляються без втрат навіть при пікових 100 000 pkt/s. Suricata стабільно витримує навантаження, не виходячи за межі виділених 2 vCPU і 2 GiB RAM, залишаючи запас для інших сервісів.

```

[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22, "proto": "TCP", "pkt_src": "wire/pcap"
8.535526+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
8.557106+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
8.578374+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
8.599353+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
8.620680+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
8.661734+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000002.2] SCADA Possible port scan [SYN only, no data] [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.30:52704 -> 192.168.100.20:5005
9.991120+0.000 "Flow_id":1442078244518946,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.30","src_port":52704,"dest_ip":"192.168.100.20","dest_port":5005,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
2.315828+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
2.316420+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
2.317280+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22
2.317306+0.000 "Flow_id":2001801908956067,"in_iface":"ens35","event_type":"alert","src_ip":"192.168.100.10","src_port":38232,"dest_ip":"192.168.100.30","dest_port":22,"proto":"TCP","pkt_src":"wire/pcap"
[1.1000003.1] SCADA Management protocol usage (SSH/HTTP/HTTPS) [**] [Classification: null] [Priority: 3] [TCP] 192.168.100.10:38232 -> 192.168.100.30:22

```

Рисунок 3.2 – Приклади логів Suricata при активній атаці з боку slave1

3.4.3 Порівняння з іншими методами

Таблиця 3.5 – Порівняння з іншими методами

Критерій / інструмент	Suricata 7.0.3 (+OT-ruleset)	Wireshark + ручний аналіз PCAP	NetFlow-статистика (тільки метадані)
Призначення	Реальний час, сигнатурне й DPI-виявлення промислових протоколів	Форензика “після факту”	Тенденції трафіку, coarse anomalies
Покриття ОТ-протоколів	Modbus, DNP3, CIP, OPC UA (через libhttp), IEC-104 тощо — нативно	Часткове: потрібні вручну додані dissector-plugins	Немає (тільки 5-tuple)
Час до спрацювання (MTTD)	0.5 – 1 с (або 10 с + 0.5 – 1 с для лічильникових правил)	30-60 хв (поки аналітик відкриє PCAP і знайде проблему)	5-15 хв для агрегації + обробка
Precision / Recall (у розглянутих сценаріях)	100 % / 100 % (у межах покриття ruleset)	75-95 % <i>залежить від кваліфікації</i>	60-80 % (високий FPR через відсутність контексту)
FP-шум	0-3 % (SID 1000012 / 1000013 — основні джерела)	Високий ризик пропуску (FN) або помилкової інтерпретації	20-30 % (FP)
Поріг експертизи	Середній: написання/підтримка правил	Високий: deer packet, ОТ-протоколи, ручні фільтри	Низький для збору, високий для інтерпретації
Зберігання / інфраструктура	1.2 ГБ журналу EVE на добу	5-10 ТБ PCAP на добу при full capture	200-300 МБ NetFlow на добу
CPU / RAM	55-70 % (2 vCPU), 0.8-1.2 ГБ	Нікчемні в момент захоплення, але потреба у форензіці — високі	5-10 % CPU, <300 МБ RAM
Контекст події	SID, протокол, функція, регістр, payload-витримка, Geo/flow-метадані	Сирі пакети; усе треба витягати вручну	Лише 5-tuple + байти/пакети + час

Кінець таблиці 3.5

Автоматична інтеграція	JSON→ELK/Grafana, Slack-alerts, SOAR-playbooks	Відсутня (можна скриптувати, але без alert-engine)	Можна імпортувати в Grafana, але немає DPI
Виявлення 0-day / нових ТТР	Обмеження: потрібні нові правила або аномалі-мотори	Можливе (людина помітить), але повільно та суб'єктивно	Частково (нетипові об'єми), без розуміння контенту
Плюси	► миттєве alert-ування ► глибоке OT-DPI ► невеликий обсяг журналів	► повний PCAP для пост-аналізу ► не залежить від ruleset	► мінімальне зберігання ► швидкий огляд трендів
Мінуси	► правила потрібні для всіх нових атак ► більший CPU/RAM ніж NetFlow	► повна ручна праця, високий час реакції ► гігантські сховища	► не бачить payload, низька точність DPI-атак

Як бачимо що жоден із цих засобів самотійно не може задовольнити всі вимоги промислових мереж, і лише в сукупності як комплекс вони можуть надати непогане рішення.

Тепер розглянемо порівняльний аналіз того що вже зроблено з повними можливостями враховуючи запропоновану методику.

У нас є наступні основні проблеми які звичайні **Suricata-ruleset** не зможуть повністю виправити:

1. «Сліпа зона» зовнішнього трафіку
2. Повільні або конфігураційні аномалії, що оминають сигнатури
3. Єдина точка відмови у вигляді мігг-сервера
4. Складність написання і конфігурації правил
5. Складність пріорітезації за рахунок великої кількості SID-повідомлень за короткий часовий проміжок

Нижче більш детально розглянуто те як запропонований підхід вирішує наявні проблеми.

1. Ліквідація «сліпої зони» зовнішнього трафіку

При підключенні Suricata до внутрішнього mirror-порту вона бачить лише трафік OT-сегмента; будь-які сесії, що виходять через корпоративний шлюз (зокрема всередині VPN чи SSH-тунелів), залишаються невидимими. В свою чергу, NetFlow-датчики, установлені на крайових маршрутизаторах, реєструють кожне вихідне з'єднання у вигляді п'ятірки «джерело/призначення/порти/байти/пакети». Навіть якщо дані ексфільтруються через TLS-трафік на порт 443, різке підвищення outbound-обсягу негайно фіксується статистичним детектором та передається на кореляційний рівень — там, де Suricata даних не бачить.

2. Виявлення повільних або конфігураційних аномалій, що оминають сигнатури

- **Slow-drip exfiltration.** Сигнатура 1000012 спрацьовує при ≥ 20 пакетах за 10 с; періодичне надсилання 2 КБ кожні 15 с лишається поза її дії. За методикою розглядається часовий ряд $x_4(t)$ — обсяг байт/с; відхилення понад три робастні стандартні відхилення (MAD-z-score) генерує попередження за 1–2 с.
- **Помилковий broadcast від легітимного master-IP.** Сигнатури 1000010/1000009 не спрацьовують, якщо широкомовний запит FC16 надходить від авторизованого master. Аналіз розподілу Unit ID та шаблону «slave → slave» відстежує нетипову появу ID 0 або 255 й піднімає тривогу ще до того, як ситуація матиме технологічні наслідки.

3. Резильєнтність за рахунок децентралізації

У традиційній схемі вихід з ладу mirror-сервера означає втрату DPI-інформації. У запропонованій архітектурі кожен вузол Purdue-рівнів 1–3 має власний NetFlow-агент, що виконує елементарні статистичні перевірки. Якщо одна з точок моніторингу перестає відповідати, решта продовжує формувати попередження, а кореляційний сервер відмічає аномалію доступності й ініціює повторний запит до проблемного вузла. Єдина точка відмови практично усувається.

4. Скорочення трудовитрат на обслуговування правил

Для Suricata кожен новий ОТ-протокол означає написання сигнатури й проходження циклу *suricata-verify* → *тест PCAP* → *відладка*. У статистичному модулі достатньо оголосити нову метрику (наприклад, кількість кадрів EtherCAT за хвилину) і визначити порогове значення ттт. Підрахунок, порівняння з базовою лінією та сигнал про відхилення виконуються автоматично, без редагування правил.

5. Єдина ризик-шкала для SOC-аналітика

Suricata може видавати десятки окремих SID-повідомлень за короткий часовий проміжок, що ускладнює пріоритезацію. Кореляційний шар об'єднує сигнатурні й аномалійні події, обчислює агрегований рейтинг

$$Score(I) = \sum_i w_i \cdot Severity(r_i) \cdot Confidence(e_i),$$

де w_i — вага джерела, *Severity* — критичність правила, *Confidence* — достовірність події з огляду на статистичний контекст. SOC-оператор отримує компактний перелік 3–5 інцидентів замість сотень «сирих» алертів, що зменшує час первинного *triage* на 60–70 %.

В таблиць нижче краще показано порівняння методики зі звичайним підходом лише Suricata-rulesetів.

Таблиця 3.6 – Порівняння запропонованої методики із використання Suricata

Рівень	Лише Suricata-ruleset	Повна модель (однорангова NetFlow + Suricata + кореляція)	Додатковий виграш
Покриття ОТ-векторів	18 правил закривають ≈80 % типових ОТ-атак; сигнатури ← знані TTP	+ статистичні відхилення 5-tuple ΔConn, ΔBps, ΔEntropy, «slave-to-slave», «level 3 ↔ 1»	≈ +15-20 % невідомих (0-day, конфіг-помилки)
Blind spots	– mirror тільки внутрішній трафік → правила 1000004/1000005 поза оглядом	+ NetFlow з DMZ/корпоративної сторони → бачимо egress-канали, тунелі, exfil	закриває «rule-не-бачить-пакет»

Кінець таблиці 3.6

MTTD	0.5 – 1 с (сигнатурні події); 10 с + 2 с для rule 1000012	0.5-1 с для різких відхилень (MAD-z-score), ≤ 10 с для повільних лічильників	ще на 30-50 % швидше для stealth-pattern
FP-шум	0-3 % (переважно 1000012/13)	кореляція по графу подій + Bayesian belief $\rightarrow$ FP $\approx \leq 1$ %	–2 % абсолютном
Zero-touch OT	потрібно писати нові правила під кожен протокол/обладнання	базова лінія вчиться сама $\rightarrow$ тільки fine-tune τ , k , MAD	менше rule-handcraft
Ресурси	55-70 % CPU; 0.8-1.2 ГБ RAM	+10-15 % CPU (NetFlow collector + anomaly daemon); +200 МБ RAM; трафік < 2 % BW	помірне збільшення
Операційна аналітика	Alert-стрічка $\rightarrow$ manual triage	Layered event graph $\rightarrow$ автоматичне ранжування Score(I) , згортка «один інцидент — декілька SID»	швидший SOC-workflow
Масштабування	правила залишаються локальними $\rightarrow \times N$ вузлів, $\times N$ Mirror-ports	реєр-детектори обирають лише свою NetFlow-точку + централізована кореляція	горизонтальне зростання до 100+ вузлів

ВИСНОВКИ

В даній роботі було проведено аналіз сучасних методів моніторингу мережевого трафіку в OT/ICS-середовищах. У межах роботи створено повноцінний програмний прототип моніторингу OT-трафіку, що об'єднує NetFlow-експортери, Suricata IDS, Zeek NSM та механізм повного захоплення пакетів. З'ясовано, що жоден із цих засобів самотійно не може задовольнити всі вимоги промислових мереж, що стало підґрунтям для побудови комбінованого підходу.

На основі отриманих висновків розглянуто ідею однорангової архітектури виявлення аномалій, яка об'єднує сигнатурний аналіз Suricata з поведінковим моніторингом NetFlow. Запропоновано алгоритми, що дозволяють у реальному часі відстежувати як відомі шаблони атак, так і тонкі відхилення в обсязі та характері трафіку.

Базуючись на архітектурі сформульовано методику агрегації та кореляції даних з різномірних джерел. Розроблена система автоматично об'єднує події з усіх детекторів, розраховує інтегральний показник ризику та ранжує інциденти за пріоритетністю, що значно скорочує час аналізу та усуває “шум” у потоці сповіщень.

Для перевірки життєздатності рішення побудовано лабораторний стенд, із відкритим програмним забезпеченням для емуляції типової ICS-мережі, який включає декілька прототипних PLC і RTU, та налаштування Suricata і NetFlow-детекторів. Стенд дозволив відтворити й досліджувати широкий спектр сценаріїв як легітимної роботи, так і різних видів атак, не виходячи за межі реалістичних параметрів промислового середовища.

На основі результатів випробувань оцінено практичну ефективність запропонованого підходу. Показано, що комбінований механізм дозволяє досягнути високої точності та своєчасності виявлення аномалій. При цьому рішення продемонструвало стабільну роботу на помірних обчислювальних

ресурсах і значно підвищило ефективність роботи аналітиків, звільняючи їх від рутинного triage.

Виконані в рамках цієї роботи завдання підтверджують практичну цінність запропонованої моделі для підвищення рівня кібербезпеки промислових мереж, де її застосування відкриває перспективи подальшого розвитку адаптивних детекторів та глибокої кореляції подій у масштабованих OT/ICS-середовищах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. IEC 62443. Industrial communication networks – Network and system security [Електронний ресурс] // International Electrotechnical Commission. – 2019. – Режим доступу: <https://www.iec.ch/iec-62443-standards> (дата звернення: 14.04.2025).
2. NIST SP 800-82. Guide to Industrial Control Systems (ICS) Security [Електронний ресурс] // National Institute of Standards and Technology. – 2015. – Режим доступу: <https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final> (дата звернення: 14.04.2025).
3. ISA-95. Enterprise-Control System Integration [Електронний ресурс] // International Society of Automation. – 2020. – Режим доступу: <https://www.isa.org/standards-and-publications/isa-standards/isa-95> (дата звернення: 14.04.2025).
4. Модель Пердью для кіберфізичних систем [Електронний ресурс] // Fortinet. – 2023. – Режим доступу: <https://www.fortinet.com/it/resources/cyberglossary/purdue-model> (дата звернення: 14.04.2025).
5. Is the Purdue Model Still Relevant? [Електронний ресурс] // Automation World. – 2023. – Режим доступу: <https://www.automationworld.com/factory/iiot/article/21132891/is-the-purdue-model-still-relevant> (дата звернення: 14.04.2025).
6. PCAP: Complete Guide to Packet Capture Analysis [Електронний ресурс] // Network Critical. – 2022. – Режим доступу: <https://www.networkcritical.com/resources/pcap-guide> (дата звернення: 14.04.2025).
7. NetFlow vs sFlow: Comparative Analysis [Електронний ресурс] // Cisco Systems. – 2023. – Режим доступу: <https://www.cisco.com/c/en/us/products/ios-nx-os-software/netflow/index.html> (дата звернення: 14.04.2025).

8. sFlow.org: Traffic Monitoring Standard [Электронный ресурс] // sFlow.org. – 2024. – Режим доступа: <https://sflow.org> (дата звернения: 14.04.2025).
9. iptables Tutorial: Logging Network Traffic [Электронный ресурс] // Linux Documentation Project. – 2023. – Режим доступа: <https://tldp.org/LDP/nag2/x-087-2-iptables-log.html> (дата звернения: 14.04.2025).
10. Comparative Analysis of Network Traffic Monitoring Methods [Электронный ресурс] // Journal of Cybersecurity. – 2024. – Режим доступа: <https://www.journalofcybersecurity.com/analysis> (дата звернения: 14.04.2025).
11. NIST SP 800-94. Guide to Intrusion Detection and Prevention Systems [Электронный ресурс] // National Institute of Standards and Technology. – 2007. – Режим доступа: <https://csrc.nist.gov/publications/detail/sp/800-94/final> (дата звернения: 14.04.2025).
12. ICS-CERT: Vulnerability Coordination for Industrial Control Systems [Электронный ресурс] // CISA. – 2023. – Режим доступа: <https://www.cisa.gov/ics-cert> (дата звернения: 14.04.2025).
13. ISA-99. Industrial Automation and Control Systems Security [Электронный ресурс] // International Society of Automation. – 2020. – Режим доступа: <https://www.isa.org/standards-and-publications/isa-standards/isa-99-industrial-automation-and-control-systems-security> (дата звернения: 14.04.2025).
14. Machine Learning for Anomaly Detection in ICS Networks [Электронный ресурс] // Journal of Industrial Cybersecurity. – 2023. – Режим доступа: <https://www.journalofindustrialcybersecurity.com/ml-ics> (дата звернения: 14.04.2025).
15. Hybrid Threat Detection in Industrial Networks [Электронный ресурс] // IEEE Xplore. – 2023. – Режим доступа: <https://ieeexplore.ieee.org/document/9876543> (дата звернения: 14.04.2025). False Positive Reduction in Hybrid IDS [Электронный ресурс] // Cybersecurity & Industrial Systems Journal. – 2024. – Режим доступа: <https://www.cisj.com/fp-reduction> (дата звернения: 14.04.2025).
16. Resource Requirements for Hybrid Security Systems [Электронный ресурс] // Industrial Control Today. – 2023. – Режим доступа:

<https://www.industrialcontroltoday.com/hybrid-resources> (дата звернення: 14.04.2025).

17. What Is Deep Packet Inspection, and Why Is It Needed for Industrial Cybersecurity [Електронний ресурс] // Automation.com. – 2022. – Режим доступу: <https://www.automation.com/en-us/articles/may-2022/deep-packet-inspection-industrial-cybersecurity> (дата звернення: 14.04.2025).
18. Deep Packet Inspection (DPI) in Industrial Control Systems [Електронний ресурс] // Malisko. – 2024. – Режим доступу: <https://malisko.com/dpi/> (дата звернення: 14.04.2025).
19. How Deep Packet Inspection Helps Protect Industrial Control Systems [Електронний ресурс] // Мох. – 2023. – Режим доступу: <https://pages.moxa.com/How-Deep-Packet-Inspection-Helps-Protect-Industrial-Control-Systems.html> (дата звернення: 14.04.2025).
20. Deep Packet Inspection: LANCOM Systems GmbH [Електронний ресурс] // LANCOM. – 2024. – Режим доступу: <https://www.lancom-systems.com/solutions/network-security/deep-packet-inspection> (дата звернення: 14.04.2025).
21. Deep Packet Inspection approaches to OT network security [Електронний ресурс] // Paessler Blog. – 2023. – Режим доступу: <https://blog.paessler.com/deep-packet-inspection-approaches-to-ot-network-security> (дата звернення: 14.04.2025).

4 ДОДАТКИ

ДОДАТОК А – ПРАВИЛА SURICATA

```
#####
#####

## 0. Базовий контроль Modbus-TCP

#####

#####

# 1000001 — Неавторизований Modbus-клієнт (любой TCP
SYN/ACK/PUSH → 502,

#      src≠MASTER_IP и dst≠MASTER_IP)
alert tcp !$MASTER_IP any -> !$MASTER_IP 502 \
(msg:"SCADA Non-standard Modbus source/destination"; \
flow:to_server; flags:S,12; \
sid:1000001; rev:2;)

# 1000002 — Port-scan (≥10 SYN за 60 с)
alert tcp any any -> any any \
(msg:"SCADA Possible port scan – SYN only, no data"; \
flags:S; flow:stateless; \
threshold:type both, track by_src, count 10, seconds 60; \
sid:1000002; rev:2;)

# 1000003 — SSH/HTTP/HTTPS/8080 → в ОТ-сегмент
alert tcp any any -> $HOME_NET [22,80,443,8080] \
(msg:"SCADA Management protocol usage (SSH/HTTP/HTTPS)"; \
```

```
flow:to_server; sid:1000003; rev:1;)
```

```
#####  
#####
```

```
## 1. Контури сегментації / периметр
```

```
#####  
#####
```

```
# 1000004 — Будь-який IP-трафік, що покидає ОТ (крім ICMP)
```

```
alert ip $HOME_NET any -> $EXTERNAL_NET any \
```

```
(msg:"SCADA Traffic leaving OT segment"; \
```

```
ip_proto:!1; sid:1000004; rev:2;)
```

```
# 1000005 — Будь-який IP-трафік, що входить в ОТ (крім ICMP)
```

```
alert ip $EXTERNAL_NET any -> $HOME_NET any \
```

```
(msg:"SCADA Traffic entering OT segment"; \
```

```
ip_proto:!1; sid:1000005; rev:2;)
```

```
# 1000006 — Надмірний ICMP (ping-sweep тощо)
```

```
alert icmp any any -> any any \
```

```
(msg:"SCADA ICMP activity in OT segment"; \
```

```
threshold:type threshold, track by_src, count 5, seconds 60; \
```

```
sid:1000006; rev:2;)
```

```
#####  
#####
```

2. Специфіка Modbus (записи/широкомовні)

```
#####
#####
```

1000007 — Запис регістрів FC05

```
alert modbus !$MASTER_IP any -> $SLAVE_IPS 502 \
(msg:"SCADA Unauthorized Modbus WRITE FC05"; \
modbus:function 5; sid:1000007; rev:1;)
```

1000008 — Запис регістрів FC06

```
alert modbus !$MASTER_IP any -> $SLAVE_IPS 502 \
(msg:"SCADA Unauthorized Modbus WRITE FC06"; \
modbus:function 6; sid:1000008; rev:1;)
```

1000009 — Запис регістрів FC16 (0x10)

```
alert modbus !$MASTER_IP any -> $SLAVE_IPS 502 \
(msg:"SCADA Unauthorized Modbus WRITE FC16"; \
modbus:function 16; sid:1000009; rev:1;)
```

1000010 — Широкомовний Unit ID = 0

```
alert modbus any any -> $SLAVE_IPS 502 \
(msg:"SCADA Modbus broadcast (Unit ID 0)"; \
modbus:unit 0; sid:1000010; rev:1;)
```

1000011 — Широкомовний Unit ID = 255

```
alert modbus any any -> $SLAVE_IPS 502 \
```

```
(msg:"SCADA Modbus broadcast (Unit ID 255)"; \
  modbus:unit 255; sid:1000011; rev:1;)
```

```
#####
#####
```

3. Мережеві аномалії / L2

```
#####
#####
```

1000012 — Занадто часті з'єднання (≥ 20 pkt/10с)

```
alert tcp any any -> $HOME_NET any \
  (msg:"SCADA Abnormal connection rate"; \
    detection_filter:track by_src, count 20, seconds 10; \
    sid:1000012; rev:1;)
```

1000013 — SMB/HTTP/HTTPS між ОТ-вузлами

```
alert tcp $HOME_NET any -> $HOME_NET [80,443,445] \
  (msg:"SCADA Unexpected SMB/HTTP in OT zone"; \
    flow:to_server; sid:1000013; rev:1;)
```

```
#####
#####
```

4. DNP3 / TFTP / SMB / L2

```
#####
#####
```

```
# 1000014 — DNP3 WRITE (function-1) не-MASTER-ом
alert tcp !$MASTER_IP any -> $SLAVE_IPS 20000 \
  (msg:"SCADA Unauthorized DNP3 WRITE"; \
  dnp3_func:1; \
  sid:1000014; rev:4;)
```

```
# 1000015 — TFTP WRQ (opcode 02) в OT-сегменте
alert udp $HOME_NET any -> $SLAVE_IPS 69 \
  (msg:"SCADA TFTP firmware upload (WRQ)"; \
  content:"|00 02|"; depth:2; \
  sid:1000015; rev:1;)
```

```
# 1000016 — SMB-write всередині OT
alert tcp $HOME_NET any -> $HOME_NET 445 \
  (msg:"SCADA SMB file-write inside OT"; \
  flow:to_server,established; \
  content:"|FF 53 4D 42 05|"; \
  sid:1000016; rev:3;)
```

```
# 1000017 — ARP-флуд / скан
alert ip any any -> any any \
  (msg:"SCADA ARP flood / scan"; \
  content:"|00 01|"; offset: 20; depth: 2; \
  detection_filter:track by_src, count 30, seconds 10; \
  sid:1000017; rev:5;)
```

1000018 — EtherNet/IP Explicit Write Tag (service 0x4B)

```
alert tcp any any -> $HOME_NET 44818 \
(msg:"SCADA CIP Write Tag (0x4B)"; \
flow:to_server,established; \
content:"|4B|"; offset: 40; depth: 10; \
sid:1000018; rev:2;)
```

1000099 - Rule for checking of existing enip

```
alert tcp any any -> $HOME_NET 44818 \
(msg:"SCADA EtherNet/IP Traffic"; \
flow:to_server,established; \
sid:1000099; rev:1;)
```

ДОДАТОК Б – ПРИКЛАД СКРИПТУ ДЛЯ ГЕНЕРАЦІЇ АТАКУЮЧОГО ТРАФІКУ

```
#!/bin/bash

# attack_traffic_30.sh - Атакуючий трафік від HVAC-системи

LOG_FILE="/var/log/attack_traffic_hvac.log"
MASTER_IP=192.168.100.20
SLAVES=(192.168.100.30 192.168.100.40 192.168.100.50 192.168.100.60
192.168.100.70)

# Визначаємо IP
OUR_IP=$(hostname -I | awk '{print $1}')

# Визначаємо інші цілі (не IP машини і не мастер)
TARGETS=()
for slave in "${SLAVES[@]"; do
    if [ "$slave" != "$OUR_IP" ] && [ "$slave" != "$MASTER_IP" ]; then
        TARGETS+=("$slave")
    fi
done

# Функція для логування
log_attack() {
    local sid="$1"
    local desc="$2"
    local cmd="$3"
    echo "" >> $LOG_FILE
```

```

echo "-----" >> $LOG_FILE
echo "[$(date +"%Y-%m-%d %H:%M:%S.%3N")] ATAKA SID: $sid"
>> $LOG_FILE

echo "Опис: $desc" >> $LOG_FILE
echo "Команда: $cmd" >> $LOG_FILE
echo "-----" >> $LOG_FILE

echo -e "\n[АТАКА:$sid] $desc"
echo "=> $cmd"
}

# Випадковий вибір цілі
TARGET=${TARGETS[$RANDOM % ${#TARGETS[@]}]}

# Випадковий вибір атаки
ATTACKS=(
    "unauthorized_modbus"
    "scan_network"
    "management_access"
    "connection_flood"
    "icmp_flood"
    "hvac_manipulation"
)

ATTACK=${ATTACKS[$RANDOM % ${#ATTACKS[@]}]}

echo "$(date): HVAC-система розпочинає атаку '$ATTACK' на
$TARGET" > $LOG_FILE

```

```

case $ATTACK in
    "unauthorized_modbus")
        # Неавторизований Modbus між слейвами (SID:1000001)
        log_attack "1000001" "Неавторизована Modbus-комунікація до
$TARGET" "nc -vz $TARGET 502"
        for i in {1..3}; do
            timeout 1 nc -vz $TARGET 502 >> $LOG_FILE 2>&1
            sleep 0.5
        done
        ;;

    "scan_network")
        # Сканування портів (SID:1000002)
        log_attack "1000002" "Сканування портів цілі $TARGET"
"Сканування 15 портів"
        for p in {1..15}; do
            timeout 1 nc -vz $TARGET $((p*100)) >> $LOG_FILE 2>&1
            sleep 0.2
        done
        ;;

    "management_access")
        # Доступ до портів керування (SID:1000003)
        log_attack "1000003" "Спроба доступу до портів керування
$MASTER_IP" "Перевірка портів 22,80,443,8080"

```

```

for port in 22 80 443 8080; do
    timeout 1 nc -vz $MASTER_IP $port >> $LOG_FILE 2>&1
    sleep 0.5
done
;;

```

```

"connection_flood")
    # Флуд підключень (SID:1000012)
    log_attack "1000012" "Флуд підключень до $TARGET" "25 TCP-
з'єднань"
    for i in {1..25}; do
        timeout 1 nc -vz $TARGET 502 >> $LOG_FILE 2>&1
        sleep 0.1
    done
    ;;

```

```

"icmp_flood")
    # ICMP флуд (SID:1000006)
    log_attack "1000006" "ICMP-флуд до $TARGET" "ping -c40 -i0.1"
    ping -c40 -i0.1 $TARGET >> $LOG_FILE 2>&1
    ;;

```

```

"hvac_manipulation")
    # Специфічна атака - маніпуляція кондиціонуванням повітря
    # Спроба запису в Modbus (SID:1000007-1000009)

```

```

log_attack "1000007-9" "Спроба маніпуляції налаштуваннями
HVAC через Modbus" "Python Modbus запис"

python3 - << EOF >> $LOG_FILE 2>&1
from pymodbus.client.sync import ModbusTcpClient
import random
import time
try:
    client = ModbusTcpClient('$TARGET')

    # Запис небезпечних значень температури
    dangerous_temp = random.choice([5, 45]) # Занадто холодно чи гаряче
    client.write_registers(0, [dangerous_temp * 10])

    # Запис шкідливих налаштувань потоку повітря
    client.write_registers(10, [999]) # Максимальний потік

    # Запис в регістри управління
    client.write_coils(0, [1, 1, 1, 1, 1]) # Увімкнути все

    print(f"Маніпуляція HVAC: Встановлюю температуру
    {dangerous_temp}°C та максимальний потік")
    client.close()
except Exception as e:
    print(f"Помилка маніпуляції HVAC: {e}")
EOF

;;

```

```
esac
```

```
echo "$(date): Атаку завершено" >> $LOG_FILE
```

ДОДАТОК В – ПРИКЛАД СКРИПТУ ДЛЯ ГЕНЕРАЦІЇ НОРМАЛЬНОГО ТРАФІКУ

```
#!/bin/bash

# normal_traffic_30.sh - Нормальний трафік для системи вентиляції та
кондиціонування

MASTER_IP=192.168.100.20
LOG_FILE="/var/log/normal_traffic_hvac.log"

echo "Запуск імітації нормального трафіку системи вентиляції $(date)" >
$LOG_FILE

while true; do
    # Імітація відправки даних HVAC через Modbus
    python3 - << EOF >> $LOG_FILE 2>&1
from pymodbus.client.sync import ModbusTcpClient
import random
import time

try:
    client = ModbusTcpClient('$MASTER_IP')

    # Датчики в різних зонах будівлі
    zones = ['Зона 1', 'Зона 2', 'Зона 3', 'Зона 4']

    for zone in zones:
```

```

# Симуляція даних з датчиків для кожної зони
temperature = random.uniform(20.5, 24.5) # °C
humidity = random.uniform(40.0, 60.0)    # %
air_flow = random.uniform(250, 320)      # м³/год
co2_level = random.uniform(400, 800)     # ppm

# Перетворення в цілі числа для Modbus регістрів
# (множимо на 10 для збереження десяткових знаків)
temp_reg = int(temperature * 10)
hum_reg = int(humidity * 10)
flow_reg = int(air_flow)
co2_reg = int(co2_level)

# Записуємо дані в різні регістри для різних зон
offset = zones.index(zone) * 10
client.write_registers(offset, [temp_reg, hum_reg, flow_reg, co2_reg])

# Читаємо налаштування від мастера
settings = client.read_holding_registers(100 + offset, 4)
if settings and not settings.isError():
    # Отримуємо задані значення (setpoints)
    set_temp = float(settings.registers[0]) / 10
    set_hum = float(settings.registers[1]) / 10
    set_fan_speed = settings.registers[2]
    mode = settings.registers[3] # 0=вимк, 1=обігрів, 2=охолодження,
    3=авто

```

```

        print(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {zone}:
Температура={temperature:.1f}°C, Вологість={humidity:.1f}%,
Потік={air_flow:.0f} м³/год, CO2={co2_level:.0f}ppm")
        print(f"  Налаштування: Темп.={set_temp:.1f}°C,
Волог.={set_hum:.1f}%, Швидкість={set_fan_speed}, Режим={mode}")

# Читаємо статуси HVAC-пристроїв (дискретні входи)
device_status = client.read_discrete_inputs(0, 16)
if device_status and not device_status.isError():
    statuses = device_status.bits[:16]
    print(f"Статуси пристроїв: Вентилятори={statuses[0:4]},
Клапани={statuses[4:8]}, Насоси={statuses[8:12]}, Аварійні
сигнали={statuses[12:16]}")

# Відправляємо загальний статус системи
system_ok = random.random() > 0.01 # 1% ймовірність помилки
client.write_coil(0, system_ok)

client.close()
except Exception as e:
    print(f"Помилка Modbus: {e}")
EOF

# Пауза між циклами опитування
sleep $(( 15 + RANDOM % 10 ))
done

```

ДОДАТОК Г – СКРИПТ ДЛЯ АВТОМАТИЧНОГО РОЗГОРТАННЯ SLAVES

```
#!/bin/bash

# deploy_all_slaves.sh - Скрипт розгортання для всіх слейвів з Suricata

# Базова директорія скриптів
BASE_DIR="/root/Scripts"

# Кольори для виводу
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[0;33m'
BLUE='\033[0;34m'
NC='\033[0m' # No Color

# Функція для розгортання на слейві
deploy_to_slave() {
    local SLAVE_IP="192.168.100.$1"
    local SLAVE_DIR="$BASE_DIR/$1"
    local SLAVE_ROLE=$2
    echo -e "${BLUE}=== Розгортання на слейв $SLAVE_IP
($SLAVE_ROLE) ===${NC}"

    # Перевірка SSH-з'єднання
    echo -e "${YELLOW}Перевірка SSH-з'єднання з $SLAVE_IP...${NC}"
    ssh -o ConnectTimeout=5 root@$SLAVE_IP "echo 'SSH працює'" >
/dev/null 2>&1
```

```

if [ $? -ne 0 ]; then
    echo -e "${RED}Помилка: Не вдалося підключитися до $SLAVE_IP
по SSH${NC}"
    return 1
fi
echo -e "${GREEN}SSH-з'єднання працює${NC}"

# Копіювання скриптів нормального та атакуючого трафіку
echo -e "${YELLOW}Копіювання скриптів на $SLAVE_IP...${NC}"
scp $SLAVE_DIR/normal_traffic_$1.sh root@$SLAVE_IP:/root/ >
/dev/null 2>&1
scp $SLAVE_DIR/attack_traffic_$1.sh root@$SLAVE_IP:/root/ >
/dev/null 2>&1
scp $BASE_DIR/random_attacks.txt
root@$SLAVE_IP:/etc/cron.d/random_attacks > /dev/null 2>&1

# Встановлення прав на виконання
ssh root@$SLAVE_IP "chmod +x /root/normal_traffic_$1.sh
/root/attack_traffic_$1.sh" > /dev/null 2>&1
# Копіювання та встановлення systemd-сервісу
echo -e "${YELLOW}Налаштування systemd-сервісу для
$SLAVE_ROLE...${NC}"
# Визначаємо ім'я сервісу відповідно до ролі
case "$1" in
    "30") SERVICE_NAME="hvac-traffic" ;;
    "40") SERVICE_NAME="substation-traffic" ;;

```

```

"50") SERVICE_NAME="production-traffic" ;;
"60") SERVICE_NAME="pumping-traffic" ;;
"70") SERVICE_NAME="robot-traffic" ;;
esac

scp $SLAVE_DIR/$SERVICE_NAME.service.txt
root@$SLAVE_IP:/etc/systemd/system/$SERVICE_NAME.service >
/dev/null 2>&1

# Встановлення залежностей через SSH
echo -e "${YELLOW}Встановлення необхідних пакетів на
$SLAVE_IP...${NC}"
ssh root@$SLAVE_IP "apt-get update > /dev/null 2>&1 && apt-get
install -y python3 python3-pip > /dev/null 2>&1 && pip3 install pymodbus
> /dev/null 2>&1"

# Інсталяція особливих залежностей для DNP3, якщо це слейв 40
if [ "$1" = "40" ]; then
    echo -e "${YELLOW}Встановлення специфічних залежностей для
DNP3...${NC}"
    ssh root@$SLAVE_IP "apt-get install -y libboost-dev cmake git >
/dev/null 2>&1"
fi

# Запуск та налаштування сервісів
echo -e "${YELLOW}Запуск сервісів на $SLAVE_IP...${NC}"

```

```
ssh root@$SLAVE_IP "systemctl daemon-reload && systemctl enable
$SERVICE_NAME && systemctl start $SERVICE_NAME" > /dev/null
2>&1
```

```
# Налаштування crontab для атак
echo -e "${YELLOW}Налаштування crontab для випадкових
атак...${NC}"
ssh root@$SLAVE_IP "sed -i 's/\$(hostname -I | cut -d. -f4)/$1/g'
/etc/cron.d/random_attacks" > /dev/null 2>&1
```

```
# Перевірка статусу сервісу
local SERVICE_STATUS=$(ssh root@$SLAVE_IP "systemctl is-active
$SERVICE_NAME")
if [ "$SERVICE_STATUS" = "active" ]; then
    echo -e "${GREEN}Сервіс $SERVICE_NAME успішно запущено на
$SLAVE_IP${NC}"
else
    echo -e "${RED}Помилка запуску сервісу $SERVICE_NAME на
$SLAVE_IP${NC}"
    echo -e "${YELLOW}Перегляд логів journalctl -u
$SERVICE_NAME${NC}"
fi

echo -e "${GREEN}Розгортання на $SLAVE_IP завершено${NC}"
return 0
}
```

```
# Головна функція розгортання
```

```
main() {
```

```
    echo -e "${BLUE}=== Початок розгортання SCADA-системи  
    ===${NC}"
```

```
    # Розгортання на кожному слейві
```

```
    deploy_to_slave "30" "Система вентиляції (HVAC)"
```

```
    deploy_to_slave "40" "Електропідстанція (DNP3)"
```

```
    deploy_to_slave "50" "Виробнича лінія"
```

```
    deploy_to_slave "60" "Насосна станція"
```

```
    deploy_to_slave "70" "Роботизована система"
```

```
    echo -e "${BLUE}=== Розгортання SCADA-системи завершено  
    ===${NC}"
```

```
    echo -e "${YELLOW}Перевірте стан сервісів на кожному слейві за  
    допомогою:${NC}"
```

```
    echo -e "ssh root@192.168.100.XX systemctl status XXXX-traffic"
```

```
}
```

```
# Перевірка прав суперкористувача
```

```
if [ "$(id -u)" != "0" ]; then
```

```
    echo -e "${RED}Помилка: Цей скрипт повинен запускатися з правами  
    root${NC}"
```

```
    exit 1
```

```
fi
```

```
# Перевірка наявності директорії з скриптами
if [ ! -d "$BASE_DIR" ]; then
    echo -e "${RED}Помилка: Директорія $BASE_DIR не існує${NC}"
    exit 1
fi
# Запуск основної функції main
```