ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи та засоби моніторингу лог-файлів у серверних і мережних інфраструктурах»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

	Віталій ЄРМОЛЕНКО
(підпис)	*Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Віталій ЄРМОЛЕНКО
Ім'я, ПРІЗВИЩЕ
Керівник: Кирило СОКОЛОВ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент:
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 Комп'ютерної інженерії
Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Єрмоленко Віталій Андрійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Методи та засоби моніторингу лог-файлів у серверних і мережних інфраструктурах
керівник кваліфікаційної роботи Кирило СОКОЛОВ, Доктор філософії,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» лютого 2026р. № 51
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна та інформаційно-довідкова література з теми моніторингу інформаційних систем, лог-менеджменту, SIEM, SOAR та застосування машинного навчання у кібербезпеці.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Огляд ефективних методів аналізу логів для виявлення інцидентів безпеки
2. Використання штучного інтелекту та машинного навчання для покращення аналізу
3. логів.
4. Метрики оцінювання ефективності моделей аналізу логів.
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	22.02-20.03.2026	Виконано
2	Обробка матеріалу	20.03-30.03.2026	Виконано
3	Розробка теоретичної частини	31.03-10.04.2026	Виконано
4	Розробка моделі мережі	11.04-20.04.2026	Виконано
5	Висновки за результатами аналізу	21.04-26.04.2026	Виконано
6	Оформлення пояснювальної записки	26.04-28.04.2026	Виконано
7	Розробка презентації	29.04-01.05.2026	Виконано
8	Передзахист	08.05-11.05.2026	Виконано
9	Здача в деканат	08.05-14.05.2026	Виконано

Здобувач вищої освіти _____ Віталій ЄРМОЛЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Кирило СОКОЛОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 84 стор., 19 рис., 4 табл., 32 джерел.

Мета роботи – Дослідження сучасних методів моніторингу та аналізу логів, а також розробка програмного застосунку для виявлення інцидентів інформаційної безпеки на основі методів машинного навчання.

Об'єктом дослідження – Процес забезпечення інформаційної безпеки в інформаційних системах.

Предметом дослідження – Методи автоматизованого аналізу логів подій для виявлення інцидентів за допомогою алгоритмів машинного навчання.

Короткий зміст роботи – Досліджено роль логів в інформаційній безпеці, типи та структуру логів, можливості моніторингу з використанням SIEM-систем; реалізовано підхід до обробки та аналізу логів за допомогою машинного навчання; побудовано модель, що успішно класифікує потенційно небезпечні події.

Отримані результати можуть бути використані при побудові або вдосконаленні систем моніторингу інформаційної безпеки на підприємствах, у тому числі з використанням SIEM-рішень або інструментів з елементами машинного навчання.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА БЕЗПЕКА, МОНІТОРИНГ ЛОГІВ, АНАЛІЗ ЛОГІВ, АНАЛІЗ ІНЦИДЕНТІВ, МАШИННЕ НАВЧАННЯ, SIEM-СИСТЕМИ, ШТУЧНИЙ ІНТЕЛЕКТ, ВИЯВЛЕННЯ АНОМАЛІЙ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ 6
ВСТУП..... 8
РОЗДІЛ 1 ОГЛЯД СИСТЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ТА ЛОГІВ 10
1.1 Поняття інформаційної безпеки та її складові 10
1.2 Логи як джерело інформації про події у системі 13
1.3 Основні типи логів та їх структура 16
Висновки до розділу 1..... 22
РОЗДІЛ 2 МЕТОДИ ТА ІНСТРУМЕНТИ МОНІТОРИНГУ ЛОГІВ 24
2.1 Методи агрегації та нормалізації даних..... 25
2.2 Моніторинг у реальному часі..... 29
2.3 Огляд SIEM-систем та його надбудову 31
Висновки до розділу 2..... 39
РОЗДІЛ 3 АНАЛІЗ ЛОГІВ ТА ВИЯВЛЕННЯ ІНЦИДЕНТІВ 40
3.1 Методи аналізу логів..... 40
3.2 Кореляція логів 41
3.3 Використання штучного інтелекту та машинного навчання для аналізу логів 44
3.4 Розпізнавання образів 48
3.5 Виявлення ефективності вашої AI або ML моделі 52
Висновки до розділу 3..... 60
РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ АНАЛІЗУ ЛОГІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ..... 62
4.1 Етапи обробки даних 63
4.2 Побудова та навчання моделі 64
4.3 Перевірка працездатності моделі..... 65
Висновки до розділу 4..... 69
ВИСНОВКИ..... 70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 72
ДОДАТОК А – КОД ПРОГРАМИ ДЛЯ ГЕНЕРУВАННЯ МОДЕЛІ 75
ДОДАТОК Б – КОД ПРОГРАММИ ДЛЯ ТЕСТУВАННЯ МОДЕЛІ..... 79
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

Active Directory (AD) – Служба каталогів Active Directory — Служба каталогів від Microsoft.

AI \ ШІ – Artificial Intelligence — Штучний інтелект.

API – Application Programming Interface — Інтерфейс програмування застосунків.

AUC-ROC – Area Under the Curve - Receiver Operating Characteristic — Площа під кривою ROC.

BYOD – Bring Your Own Device — Використання особистих пристроїв на роботі.

CEF – Common Event Format — Стандартний формат подій.

CSV – Comma-Separated Values — Значення, розділені комами.

DLL – Dynamic Link Library — Бібліотека динамічного компонування.

FPR – False Positive Rate — Частка хибно позитивних.

IDS/IPS – Intrusion Detection System / Intrusion Prevention System — Система виявлення вторгнень / Система запобігання вторгненням.

IoT – Internet of Things — Інтернет речей.

IR – Incident Response — Реагування на інциденти.

JSON – JavaScript Object Notation — Нотація об'єктів JavaScript.

ML \ МН – Machine Learning — Машинне навчання.

MSSP – Managed Security Service Provider — Постачальник керованих послуг безпеки.

SEM – Security Event Management — Управління подіями безпеки.

SIEM – Security Information and Event Management — Управління інформацією та подіями безпеки.

SIM – Security Information Management — Управління інформацією безпеки.

SOAR – Security Orchestration, Automation and Response — Оркестрація, автоматизація та реагування на інциденти безпеки.

SOC – Security Operations Center — Центр операцій безпеки.

TPR – True Positive Rate — Частка істинно позитивних.

UEBA – User and Entity Behavior Analytics — Аналітика поведінки користувачів та об'єктів.

UI/UX – User Interface / User Experience — Інтерфейс користувача / Досвід користувача.

VPN – Virtual Private Network — Віртуальна приватна мережа.

XML – eXtensible Markup Language — Розширювана мова розмітки.

ВСТУП

У сучасному світі інформаційна безпека набуває все більшого значення, оскільки кібератаки стають складнішими й масштабнішими. Логи інформаційних систем — важливе джерело даних для виявлення підозрілих активностей і запобігання інцидентам безпеки. Проте обробка логів у реальному часі є складним завданням, яке потребує спеціалізованих методів та інструментів.

Актуальність дослідження полягає у необхідності вдосконалення процесів моніторингу логів задля своєчасного виявлення загроз. Сучасні рішення, зокрема SIEM-системи та використання методів ШІ, дозволяють підвищити ефективність моніторингу безпеки, однак існує потреба в подальшому удосконаленні підходів до обробки логів, особливо з урахуванням специфіки кібербезпеки в Україні.

Таким чином, робота спрямована на вивчення сучасних методів моніторингу та аналізу логів для покращення систем інформаційної безпеки.

Мета роботи полягає у дослідженні сучасних методів і засобів моніторингу та аналізу логів для підвищення ефективності виявлення інцидентів ІБ.

Основні завдання дослідження охоплюють огляд систем інформаційної безпеки та аналіз типів логів, які вони формують. Розглянуто методи моніторингу логів у реальному часі та з певною періодичністю. Особливу увагу приділено принципам роботи SIEM-систем, що забезпечують централізоване збирання, зберігання та аналіз логів. Проаналізовано методи обробки логів із використанням кореляції для виявлення загроз. Окремо розглянуто можливості машинного навчання у виявленні інцидентів інформаційної безпеки.

Об'єкт дослідження — процеси моніторингу та аналізу логів в інформаційних системах.

Предмет дослідження — методи обробки логів, системи моніторингу безпеки та алгоритми виявлення інцидентів.

Результати роботи можуть бути використані для оптимізації процесів моніторингу інформаційних систем, підвищення швидкості та точності виявлення інцидентів безпеки.

Розділ 1 ОГЛЯД СИСТЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ТА ЛОГІВ

1.1 Поняття інформаційної безпеки та її складові

Інформаційна безпека (InfoSec) – це набір процедур та інструментів, які захищають усю корпоративну інформацію від неправомірного використання, несанкціонованого доступу, псування або знищення. InfoSec включає безпеку на фізичному й корпоративному рівні, керування доступом і кібербезпеку, так звану інформаційну тріаду (C.I.A. Triad), яка зображена на (Рис.1.1).

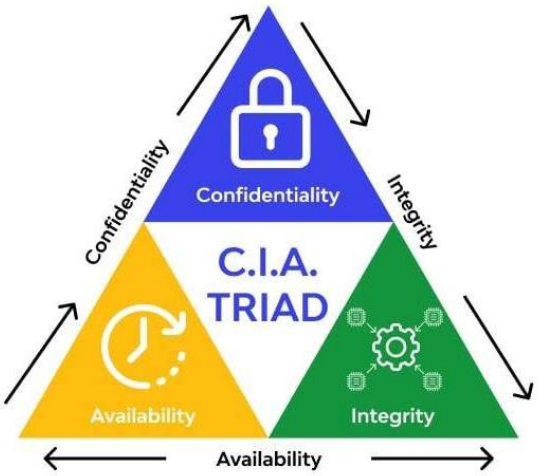


Рис. 1.1. Інформаційний трикутник [22]

Це рішення часто передбачає використання таких технологій, як посередники, що забезпечують захищений доступ до хмари, інструменти для виявлення обману, протидія загрозам на кінцевих точках, перевірка системи безпеки.

InfoSec включає низку інструментів, рішень і процесів, які захищають корпоративну інформацію на різних пристроях та в різних розташуваннях, а також допомагають убезпечитися від кібератак або інших порушень таких, як:

- Захист програм: політики, процедури, інструменти й практичні поради щодо захисту програм і даних, які містяться в них.

- Безпека в хмарі: політики, процедури, інструменти й практичні поради щодо комплексного захисту хмари, зокрема систем, даних, програм та інфраструктури.

- Криптографія: заснований на алгоритмі метод забезпечення захищеного спілкування, який полягає в тому, що певне повідомлення можуть переглядати та дешифрувати лише конкретні одержувачі.

- Аварійне відновлення: метод відновлення функціональних технологічних систем після стихійних лих, кібератак або інших порушень.

- Реагування на інциденти: план організації для реагування на кібератаки, порушення безпеки даних та інші загрози, керування ними й усунення їхніх наслідків.

- Убезпечення інфраструктури: безпека всієї технологічної інфраструктури організації, зокрема систем апаратного й програмного забезпечення.

- Керування вразливостями: процес виявлення, оцінювання й усунення вразливостей у кінцевих точках, програмному забезпеченні та системах організації.

Конфіденційність, цілісність і доступність – це запорука ефективного захисту даних та безпеки інфраструктури організації. Ці три поняття основоположні принципи для впровадження плану InfoSec.

InfoSec включає безперервну підтримку фізичного обладнання й регулярне оновлення системи, щоб авторизовані користувачі мали надійний і узгоджений доступ до даних відповідно до своїх потреб.

- Атака удосконаленої постійної загрози - це складна кібератака, яка полягає в тому, що зловмисник (або група зловмисників) отримує доступ до корпоративної мережі та даних і залишається непоміченим протягом тривалого періоду.

- Бот-мережа термін утворений від словосполучення "robot network" (роботизована мережа) і означає мережу пов’язаних пристроїв, які кіберзлочинець уражає зловмисним кодом, а потім віддалено керує ними.

- Розподілена атака на відмову в обслуговуванні (DDoS-атака) – DDoS-атаки виконуються за допомогою бот-мереж і полягають у заспамленні корпоративних вебсайтів та програм, у результаті чого для справжніх користувачів служба стає недоступною через її несправність чи відмову в обслуговуванні.

- Атака тіньового завантаження – це фрагмент зловмисного коду, який автоматично завантажується на пристрій користувача після того, як той відвідує вебсайт. Таким чином користувач стає вразливим до подальших кіберзагроз.

- Набір експлойтів – це комплексний набір інструментів, які використовують експлойти для виявлення вразливостей і враження пристроїв шкідливим програмним забезпеченням.

- Внутрішня загроза – це імовірність того, що працівник організації навмисно або випадково використовуватиме авторизований доступ, що зашкодить корпоративним системам, мережам і даним або зробить їх уразливими.

- Атака зловмисного посередника – злочинець перериває лінію зв’язку або передавання даних, маскуючись під дійсного користувача, щоб викрадати інформацію.

- Фішингова атака – під час фішингових атак зловмисники маскуються під справжні організації або дійсних користувачів, щоб викрадати інформацію через електронні листи, текстові повідомлення або інші методи спілкування.

- Зловмисна програма з вимогою викупу – це атака зловмисних програм, яка шифрує інформацію цілої організації або окремих користувачів, блокуючи до них доступ, доки не буде сплачено викуп.

- Соціотехніка – цей тип кібератак включає людський фактор. Він полягає в тому, що зловмисник завойовує довіру жертви, використовуючи байтинг, підробні антивірусні програми або фішинг, а потім збирає персональні дані та використовує їх для подальших атак.

- Атака в соцмережах – під час таких атак кіберзлочинці вражають платформи соціальних мереж, використовуючи їх як механізми поширення шкідливого програмного забезпечення, або крадуть інформацію чи дані користувачів.
- Віруси та хробаки – це приховане шкідливе програмне забезпечення, яке може самостійно поширюватися в мережі або системі користувачів.[1]

Отже, інформаційна безпека є невід'ємною складовою сучасної цифрової інфраструктури. Вона охоплює широкий спектр заходів, інструментів і принципів, спрямованих на забезпечення конфіденційності, цілісності та доступності даних. Ефективне впровадження InfoSec дозволяє організаціям протистояти численним загрозам — від фішингових атак до складних вторгнень, — мінімізуючи ризики втрати або викрадення інформації. Сучасні технології, такі як криптографія, захист хмари, реагування на інциденти та поведінкова аналітика, є критично важливими для побудови надійної системи безпеки. У світі, де обсяг і цінність цифрових даних постійно зростають, InfoSec є запорукою стабільної роботи організацій і довіри користувачів.

1.2 Логи як джерело інформації про події у системі

Логи (журнали подій) — це запис подій, що відбуваються в системах і мережах організації. Логи складаються із записів журналу, кожен запис містить інформацію, пов'язану з конкретною подією, що сталася в системі або мережі.

Багато логів в організації містять записи, пов'язані із комп'ютерної безпеки. Ці журнали безпеки комп'ютера генеруються багатьма джерелами, зокрема:

- програмним забезпеченням безпеки, наприклад, антивірусне програмне забезпечення, брандмауери та системи виявлення та запобігання вторгненням IDS/IPS;

- операційні системи на сервері, робочі станції, маршрутизатори, комутатори, мережеве обладнання, програми (Syslog, Fail2ban, Suricata)
- різними застосунками (Apache, MySQL, PostgreSQL) або операційними системами (Windows Event Logs, Linux Syslog)

Кількість і різноманітність журналів комп'ютерної безпеки суттєво зросли, що призвело до появи різних конфігурацій логів — процесів створення, зберігання, аналізу та знищення даних безпеки. Ефективне керування журналами забезпечує наявність детальних записів упродовж необхідного часу.

Рекомендації щодо зберігання, аналізу та інших аспектів логів наведені у таблиці 1.1.

Таблиця 1.1. – Приклади налаштувань конфігурації логів [21]

Категорія	**Системи з низьким впливом**	**Системи з помірним впливом**	**Системи з високим впливом**
Як довго зберігати дані журналу	1–2 тижні	1–3 місяці	3–12 місяців
Як часто проводити ротацію журналів	За бажанням (якщо виконано, принаймні раз на тиждень або кожні 25 МБ)	Кожні 6–24 години або кожні 2–5 МБ	Кожні 15–60 хвилин або кожні 0,5–1,0 МБ
Як часто система повинна передавати дані журналу до інфраструктури управління журналами	Кожні 3–24 години	Кожні 15–60 хвилин	Принаймні кожні 5 хвилин
Як часто необхідно аналізувати дані журналу локально (автоматично або вручну)	Кожні 1–7 днів	Кожні 12–24 години	Принаймні 6 разів на день
Чи потрібно перевіряти цілісність змінених логів	За бажанням	Так	Так

У разі виникнення помилки системний адміністратор або представник служби підтримки повинен визначити причину помилки, спробувати відновити втрачені дані та запобігти повторенню помилки. Це корисно, якщо програми, операційна система та інші системні служби записують важливі події, такі як брак пам’яті або надмірні спроби доступу до диска. Після цього системний адміністратор може використовувати журнал подій, щоб визначити, які умови спричинили помилку, і визначити контекст, у якому вона сталася. Періодично переглядаючи журнал подій, системний адміністратор може виявити проблеми (наприклад, несправність жорсткого диска) до того, як вони спричинять пошкодження.[3]

Існує п’ять типів подій, які можна залогувати. Усі вони мають чітко визначені загальні дані та, за бажанням, можуть включати дані про конкретні події.

Програма вказує тип події, коли повідомляє про подію. Кожна подія має бути одного типу. Засіб перегляду подій відображає окрему піктограму для кожного типу в списку журналу подій. У наведеній Таблиці 1.2. описано п’ять типів подій, які використовуються в журналі подій. [4]

Таблиця 1.2. – Вид події та її опис

Тип події	**Опис**
Помилка	Подія, яка вказує на значну проблему, наприклад втрату даних або втрату функціональності. Наприклад, якщо служба не завантажується під час запуску, реєструється подія Error.
УВАГА	Подія, яка не обов’язково є важливою, але може вказувати на можливу проблему в майбутньому. Наприклад, коли на диску мало місця, реєструється подія Попередження. Якщо програма може відновити подію без втрати функціональності або даних, вона може класифікувати подію як подію попередження.

Продовження таблиці 1.2

Тип події	Опис
Інформація	Подія, яка описує успішну роботу програми, драйвера або служби. Наприклад, коли мережевий драйвер успішно завантажується, може бути доречним зареєструвати інформаційну подію. Зауважте, що для програми для настільних комп'ютерів неприйнятно реєструвати подію щоразу під час її запуску.
Аудит успіху	Подія, яка фіксує перевірену успішну спробу безпечного доступу. Наприклад, успішна спроба користувача увійти в систему реєструється як подія аудиту успіху.

Отже логи є ключовим інструментом для моніторингу, аналізу та підтримки безпеки інформаційних систем. Вони фіксують усі значущі події, що відбуваються в інфраструктурі організації — від звичайного запуску програм до критичних помилок і спроб несанкціонованого доступу. Правильне керування логами забезпечує можливість вчасно виявляти загрози, проводити розслідування інцидентів, усувати збої та дотримуватись вимог нормативно-правових актів. Визначення типів подій, їх регулярний аналіз і збереження в структурованому вигляді дозволяє системним адміністраторам не лише реагувати на помилки, а й запобігати серйозним збоям у майбутньому. У сучасних умовах логи є не лише джерелом інформації, а й важливим компонентом системної безпеки.

1.3 Основні типи логів та їх структура

Логування в кібербезпеці відіграє важливу роль у забезпеченні захисту інформації, оскільки організації прагнуть захистити свої цінні дані та системи. Проактивний підхід є ключовим, і одним із найважливіших елементів цього підходу є ефективне використання можливостей систем безпеки інформації та керування подіями (SIEM), що сприяє своєчасному виявленню загроз і реагуванню на

інциденти. Для забезпечення належного рівня безпеки важливо визначити, які саме журнали подій слід відстежувати. До основних видів логів, що відіграють критичну роль у кібербезпеці, належать такі:

- Логи пристрою периметра (Perimeter device logs)

Пристрої периметра відстежують і регулюють трафік до та з мережі. Брандмауери, віртуальні приватні мережі (VPN), системи виявлення вторгнень (IDS) і системи запобігання вторгненням (IPS) є одними з пристроїв периметра. Ці пристрої генерують журнали, що містять велику кількість даних, а журнали пристроїв периметра життєво важливі для розуміння подій безпеки, що відбуваються в мережі. Дані у форматі системного логу допомагають IT-адміністраторам виконувати перевірки безпеки, вирішувати проблеми з роботою та краще розуміти трафік, що проходить через корпоративну мережу та з неї. Контроль даних журналу пристроїв периметра важливий, оскільки він дозволяє:

- Виявляти зловмисний трафік, IP-адреси та спроби несанкціонованого доступу;
- Сприяє знаходженню помилок конфігурацій, які можуть призвести до вразливостей;
- Для виявлення атак: Наприклад, коли сервер протягом короткого часу отримує велику кількість SYN-пакетів для підключення до сервера, це може означати розподілену атаку типу «відмова в обслуговуванні» (DDoS).

Запис на (Рис.1.2). представлено приклад логів брандмауера з міткою часу, IP-адресами, портами та протоколом. Такий запис допомагає визначити підозрілу активність, наприклад спроби підключення до незадіяних портів. Завдяки системному аналізу подібних записів, аналітики можуть виявити загрози на ранніх етапах та вжити відповідних заходів.

```
    2015-07-06 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63064
135 0 - 0 0 0 - - - SEND
```

Рис. 1.2. Розбір даних журналу типового пристрою брандмауера.

- Логи подій Windows (Windows event logs)

Журнали подій Windows — це записи всього, що відбувається в системі Windows. Ці дані журналу далі класифікуються на:

- Логи програм – фіксують збої у додатках;
- Логи безпеки – події входу, видалення файлів, спроби доступу;
- Системні логи – описують роботу операційної системи;
- Логи каталогів (AD) – автентифікація, зміни прав користувачів;
- Логи DNS-сервера – IP-клієнтів, запитувані домени тощо.

Навіщо потрібно відстежувати журнали подій Windows?

- Для забезпечення цілісності та захисту серверів і критичних ресурсів;
- Для моніторингу користувацької активності, що важливо при розслідуванні інцидентів;
- Для діагностики апаратних проблем на робочих станціях.

Windows класифікує кожну подію на основі її серйозності як попередження, інформацію, критичну та помилку. Рівень безпеки випадку в лозі що наведений на (Рис.1.3). наведено приклад логу зі служби WLAN AutoConfig, яка фіксує обмежений доступ до мережі. Такі журнали дозволяють SIEM-системам аналізувати подібні ситуації на різних пристроях у реальному часі, виявляти тренди проблем з підключенням і автоматизувати реагування.

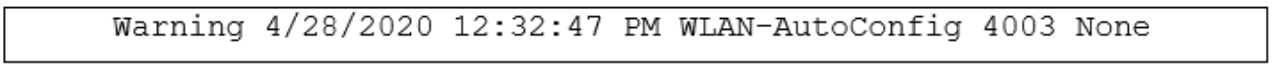


Рис. 1.3. Розбір типового журналу подій Windows

- Логи кінцевих точок (Endpoint logs)

Кінцеві точки – це пристрої, які підключені через мережу та обмінюються даними з іншими пристроями через сервери. Деякі приклади включають настільні комп’ютери, ноутбуки, смартфони та принтери. Оскільки організації все частіше

використовують віддалену роботу, кінцеві точки створюють точки входу в мережу, якими можуть скористатися зловмисники.

Моніторинг логів кінцевих точок дозволяє:

- виявляти використання знімних дисків для копіювання або перенесення потенційно шкідливих файлів;
- забезпечувати відповідність корпоративним політикам безпеки;
- аналізувати проблеми, що виникають у користувачів, на технічному рівні.

Лог що наведенний на (Рис.1.4) показує помилку драйвера Easy Print, яка може завадити користувачеві надрукувати файл. Аналіз подібних логів дозволяє не тільки усунути конкретну проблему, але й ідентифікувати серійні збої, які вимагають оновлення драйверів або змін у конфігурації системи.

```
     Error 6/20/2019 5:00:45 PM Terminal Services- Printers 1111
None
```

Рис. 1.4. Розбір типового журналу кінцевих пристроїв

- Логи додатків (Application logs)

Сучасні компанії використовують безліч додатків — від баз даних до веб-серверів та спеціалізованих програм. Усі ці програми створюють журнали, які відображають дії користувачів, системні повідомлення, помилки та інші події.Навіщо вам потрібно контролювати журнали програм?

- Щоб усунути проблеми: ці журнали допомагають виявити та виправити проблеми, пов’язані з продуктивністю та безпекою програм.
- Для моніторингу активності: журнали, згенеровані з бази даних, вказують на запити від користувачів. Це можна використовувати для виявлення несанкціонованого доступу або підозрілі SQL-запити.

Наведений приклад на (Рис.1.5) показує лог бази даних Oracle із зафіксованим підключенням, що містить дату, час, IP-адресу клієнта, ім’я користувача й порт. Такі

дані допомагають ідентифікувати підозрілі з’єднання та реагувати на інцидентиідентифікувати підозрілі з’єднання та реагувати на інциденти.

```
     02-AUG-2013  17:38:48  *  (CONNECT_DATA=(SERVICE_NAME=dev12c)
(CID= (PROGRAM=sqlplus)(HOST=oralinux1)(USER=oracle))) * (ADDRESS=
(PROTOCOL=tcp)(HOST=192.168.2.121)(PORT=21165))   *   establish   *
dev12c * 0
```

Рис. 1.5. Розбір типового журналу програми

- Проксі логи (Proxy logs)

Проксі-сервери є проміжною ланкою між внутрішньою мережею компанії та Інтернетом. Вони виконують функції фільтрації, кешування та контролю доступу. Всі запити до зовнішніх ресурсів проходять через проксі, тому його журнали мають високу аналітичну цінність.

Навіщо вам потрібно контролювати журнали проксі?

- Базовий рівень поведінки користувачів: аналіз дій користувачів у веб-переглядачі за зібраними журналами проксі-сервера може допомогти сформувати базовий рівень їхньої поведінки. Будь-яке відхилення від базової лінії може виявити порушення даних і вказати на необхідність подальшої перевірки.
- Щоб контролювати довжину пакетів: фіксують обсяг і розмір трафіку що допомагає відстежити великі або повторювані пакети даних, характерні для шкідливих дій (наприклад, ботнет-трафік або C2-комунікації).

Лог на (Рис.1.6) вказує, що User-001 зробив запити до Wikipedia, а також фіксується час, дата й URL-адреси. Аналіз запитів, URL-адрес і позначок часу в журналах допомагає виявити закономірності та допомагає у відновленні доказів у разі події.

```
     4/8/2020    2:20:55    PM    User-001    192.168.10.10    GET
https://wikipedia.com/
```

Рис. 1.6. Розбір типового журналу проксі

- Логи IoT (IoT logs)

Інтернет речей (IoT) — це мережа фізичних пристроїв з вбудованими датчиками, процесорами та програмним забезпеченням, які обмінюються даними через Інтернет. Як і інші кінцеві точки, IoT-пристрої генерують журнали, що відображають роботу мікроконтролерів, потребу в оновленнях мікропрограми та потоки даних.

Оскільки ці пристрої мають обмежену пам’ять, журнали пересилаються до централізованих систем для зберігання й аналізу. Там, зокрема в системах SIEM, відбувається обробка даних з метою виявлення збоїв і загроз.

Журнали з різних джерел (включно з IoT) об'єднуються у централізованих рішеннях, де корелюються та аналізуються задля створення повної картини стану безпеки мережі.

Формати журналів можуть бути різними:.

CSV — це формат файлу, який зберігає значення у форматі, розділеному комами. Це формат звичайного текстового файлу, який дозволяє легко імпортувати файли CSV у базу даних, незалежно від використовуваного програмного забезпечення. Оскільки файли CSV не є ієрархічними чи об’єктно-орієнтованими, їх також легше конвертувати в інші типи файлів.

JavaScript Object Notation (JSON) — це текстовий формат для зберігання даних. Це структурований формат, який полегшує аналіз збережених журналів. Його також можна запитувати для певних полів. Ці додаткові функції роблять JSON дуже надійним форматом для керування журналами. [5]

Такий аналіз допомагає виявити шаблони, аномалії та загрози, що дозволяє IT-фахівцям ухвалювати обґрунтовані рішення для підтримки надійної й безпечної інфраструктури.

Розуміння структури та особливостей різних типів журналів важливий для ефективного моніторингу подій і швидкого реагування на кіберзагрози. Кожен тип логів (периметрові пристрої, події Windows, кінцеві точки, додатки, проксі та IoT) надає цінну інформацію про стан IT-системи. Їх об'єднання в межах SIEM-рішень

створює цілісну картину безпеки, дозволяє виявити потенційні загрози і запобігти шкоді ще до виникнення інциденту.

Висновки до розділу 1

Логування є невіддільною складовою сучасної системи кібербезпеки. У процесі функціонування будь-якої інформаційної системи виникає безліч подій, які потребують реєстрації, збереження та аналізу. Саме для цього використовуються логи — структуровані записи, які містять детальну інформацію про активність користувачів, системних компонентів, мережевого обладнання та програмного забезпечення.

Існує велика кількість типів логів, кожен з яких виконує свою специфічну роль у загальному процесі моніторингу та реагування на інциденти. Наприклад, логи подій Windows дозволяють відстежувати активність у середовищі операційної системи, лог-файли кінцевих точок фіксують поведінку користувачів та програм, мережеві логи допомагають виявити підозрілий трафік, а журнали безпеки фіксують події, пов’язані з можливими загрозами. Логи додатків забезпечують контроль за функціонуванням програмного забезпечення, а логи зовнішніх пристроїв та хмарних сервісів дозволяють забезпечити цілісність та безпеку взаємодії з зовнішнім середовищем.

Важливим є те, що структура логів повинна бути логічною, послідовною та стандартизованою, аби їх можна було ефективно обробляти та аналізувати. Типовими елементами логів є дата й час події, тип події, джерело, користувач або процес, що її ініціював, а також інші технічні деталі, які дозволяють відновити хід подій.

Завдяки використанню систем управління інформацією та подіями безпеки (SIEM) логи можуть оброблятись автоматично, що значно підвищує ефективність виявлення та реагування на інциденти. Аналітики отримують змогу швидко

ідентифікувати підозрілу активність, локалізувати джерело загрози, а також зібрати докази для подальшого розслідування чи судового розгляду.

Розділ 2 МЕТОДИ ТА ІНСТРУМЕНТИ МОНІТОРИНГУ ЛОГІВ

Моніторинг журналів — це практика ретельного аналізу даних журналів, зібраних із різних компонентів вашого бізнес-середовища, включаючи веб-сервіси, хмарні платформи, бази даних, мережеве обладнання тощо, і реагування на них. Завдяки постійному аналізу та візуалізації даних журналу ви можете швидко виявляти незвичайну активність, ризики безпеці, системні збої та вузькі місця продуктивності, що сприяє спостережливості та дозволяє набагато швидше реагувати на інциденти.

Сучасні програмні системи неймовірно складні, з незліченною кількістю взаємопов’язаних частин, які генерують гори журнальних даних. Намагатися розібратися в цій інформації без сторонньої допомоги — це все одно, що шукати голку в стозі сіна. Саме тому моніторинг журналів є таким критичним. Завдяки системі моніторингу журналів ви зможете бачити те, що відбувається в реальному часі. Автоматичні сповіщення миттєво визначають проблеми, дозволяючи швидко й ефективно реагувати.

Але моніторинг логів полягає не лише у вирішенні проблем після їх виникнення; це також перш за все про запобігання їм. Виявляючи ознаки раннього попередження, ви можете завчасно вирішувати проблеми та підтримувати безперебійну роботу своїх систем.

Моніторинг журналів — це багатоетапний процес, який починається з визначення відповідних джерел журналів. Тут ви точно визначаєте системи та програми, які генерують відповідні дані журналу для ваших цілей моніторингу.

Далі йде агрегація журналів. Замість того, щоб мати журнали, розкидані по різних місцях, ви об’єднуєте їх у центральне сховище за допомогою розсилувачів журналів. Агрегація журналів часто супроводжується синтаксичним аналізом журналів, коли необроблені записи журналу нормалізуються в узгоджений

структурований формат і збагачуються відповідною контекстною інформацією. Це важливий крок для полегшення аналізу та порівняння журналів.

Серце моніторингу журналів полягає в аналізі та візуалізації даних журналу. Обрана вами платформа для керування журналами має забезпечувати можливість аналізу журналів, виконувати складні запити та виявляти приховані шаблони та аномалії. Візуальні інформаційні панелі оживляють ці дані, спрощуючи визначення тенденцій і передачу інформації. Побачити як відбувається візуалізація логів можна на (Рис.2.1). [16]

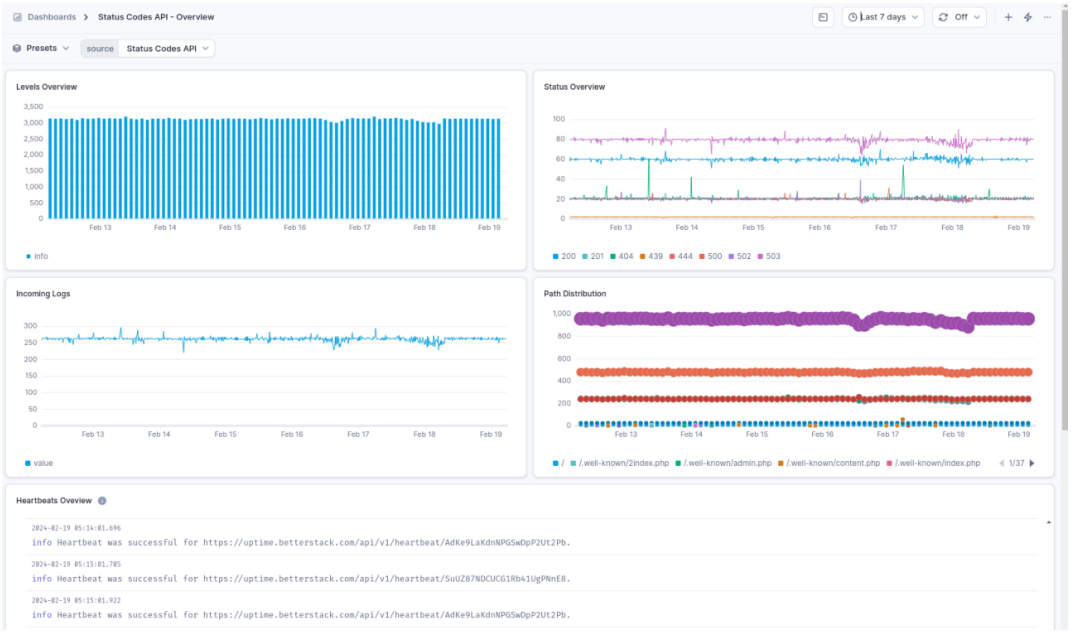


Рис. 2.1. Візуалізація логів які були зібрані протягом семи днів

2.1 Методи агрегації та нормалізації даних

2.1.1 Агрегація даних.

Агрегація даних — це процес збору, узгодження та інтеграції інформації з різних джерел для створення єдиного, узагальненого представлення даних. Такий підхід дозволяє організаціям здійснювати ефективний аналіз, формувати звітність і приймати рішення на основі повної та структурованої інформації.

Процес агрегації включає кілька ключових етапів:

1. Збір даних. Інформація надходить із внутрішніх систем (продажі, CRM, веб-аналітика) або зовнішніх джерел (API, сторонні провайдери). На цьому етапі важливо забезпечити повноту та актуальність даних.

2. Інтеграція. Отримані дані узгоджуються за структурою та форматом. Виявляються та усуваються дублікати, невідповідності та помилки. Всі джерела приводяться до єдиної моделі даних, що дозволяє уникнути суперечностей.

3. Зберігання. Агреговані дані розміщуються в централізованому сховищі — структурованому або напівструктурованому, залежно від типу інформації. Це сховище виступає єдиним джерелом правди (single source of truth) для подальшого аналізу.

4. Обробка та аналіз. Зібрані дані піддаються обробці з використанням статистичних методів, машинного навчання або інструментів візуалізації. Мета — виявлення закономірностей, аномалій, кореляцій та інших інсайтів, які можна використати для прийняття рішень.

5. Звітність. Результати аналізу подаються у формі звітів, дашбордів або графіків, що дозволяє користувачам інтерпретувати інформацію швидко та ефективно.

Агрегація даних забезпечує низку практичних переваг. Вона сприяє оцінці ефективності діяльності компанії, дозволяє контролювати ключові показники (KPI), виявляти ризики та прогнозувати тенденції. У фінансовому секторі, наприклад, агреговані дані застосовуються для виявлення шахрайства, оцінки кредитоспроможності та аналізу ринкових ризиків.

Попри переваги, агрегація даних супроводжується низкою викликів:

- Якість даних. Неточні або неактуальні дані можуть спричинити хибні висновки. Необхідно впроваджувати механізми валідації, очищення та моніторингу якості.
- Конфіденційність і безпека. Обробка персональних даних вимагає дотримання нормативних вимог (наприклад, GDPR). Організаціям потрібно забезпечити надійне шифрування, контроль доступу та регулярні аудити безпеки.

- Управління даними. Важливо чітко визначити відповідальних за дані, впровадити політики, забезпечити прозорість процесів інтеграції та обробки.
- Масштабованість. Обсяги даних зростають експоненційно, тому інфраструктура має бути готовою до роботи з великими даними. Для цього застосовуються хмарні рішення, розподілені системи обробки та високопродуктивні СУБД.

Таким чином, агрегація даних є невід'ємною складовою сучасного управління інформацією. За умови правильної реалізації вона забезпечує конкурентні переваги, підвищує якість управлінських рішень та дозволяє ефективно адаптуватися до змін у середовищі, орієнтованому на дані.[11]

2.1.2 Нормалізація логів

Нормалізація журналів (логів) — це процес перетворення даних у єдиний, стандартизований формат, що забезпечує їхню узгодженість та зручність для аналізу. Всі події на пристроях, серверах, у базах даних і додатках фіксуються в логах, які без нормалізації можуть мати різну структуру та формат.

Під час нормалізації кожне поле журналу даних приводиться до єдиного формату наприклад, дати, IP-адреси або коди подій. Також відбувається класифікація подій за категоріями. Це робиться для того, щоб важливі деталі не залишились поза увагою. У таблиці 2.1. наведено приклад, як змінюється структура лога у результаті нормалізації — значення вирівнюються, уніфікуються часові мітки та позначки рівня критичності.

Таблиця 2.1. – Відображення лога до та після нормалізації

Відображення логу до обробки	`127.0.0.1 user-identifier andy [12/Nov/2021:20:25:11-0700] "GET /apache_pb.gif HTTP/1.0"`

Продовження таблиці 2.1

Відображення логу після обробки	"127.0.0.1" refers to the client's IP address when a server request is made. "user-identifier" refers to the client's identification protocol. "andy" refers to the user ID. "[12/Nov/2021:20:25:11-0700]" refers to the date and time the

Наявність єдиного формату значно полегшує не лише аналіз, але й звітування, особливо в умовах роботи з різнорідними системами. Проте нормалізація логів — це доволі ресурсомісткий процес, особливо при великому обсязі даних або складній структурі записів.

Існує кілька поширених форматів логів, які можуть використовуватись у процесі нормалізації:

- CSV (Comma-Separated Values) — простий текстовий формат, у якому дані розділені комами. Підходить для зведених, детальних журналів і журналів помилок. Легко конвертується в інші формати.
- JSON — сучасний структурований формат, який дає змогу швидко звертатися до окремих полів. Добре підходить для автоматичного аналізу і взаємодії з API.
- Syslog — складається з трьох основних елементів: коду пріоритету, імені хоста та повідомлення. Більш зручний для читання людиною, а не машиною.
- XML — формат із чіткою структурою на основі тегів, забезпечує гнучкість і інтеграцію з зовнішніми системами.
- CEF (Common Event Format) — формат, що складається з заголовка та розширення у вигляді пар ключ-значення. Широко використовується у системах безпеки та сумісний із багатьма SIEM-платформами.

У процесі нормалізації використовуються аналізатори, які витягують ключову інформацію з «сирих» логів, розбиваючи їх на структуровані елементи. Це дозволяє:

- Створити єдину модель даних із логів, що надходять із різних джерел.
- Спрощено класифікувати події за критичністю або джерелом.

- Прискорити пошук ефективної моделі машинного навчання для подальшого аналізу.
- Виявити підозрілу або нерегулярну мережеву активність.

Крім цього, нормалізовані логи істотно полегшують діагностику — технічний персонал може оперативно виявляти вузькі місця, проблеми з продуктивністю або збої. Однак, ефективне управління логами вимагає значних зусиль, налаштувань і контролю, що підкреслює важливість комплексного підходу.[12]

У підсумку, агрегація та нормалізація логів є ключовими елементами сучасної системи моніторингу. Агрегація дозволяє зібрати фрагментовані дані в єдину структуру, а нормалізація забезпечує їхню обробку та уніфікацію. Разом вони створюють надійну основу для аналітики, підвищують точність, знижують ризики та забезпечують швидке реагування на інциденти.

2.2 Моніторинг у реальному часі

Моніторинг у реальному часі — це процес безперервного збору, обробки та аналізу даних із мінімальною затримкою. Завдяки цьому дані майже миттєво доступні для прийняття рішень, що є критично важливим у динамічному IT-середовищі. Такий підхід дозволяє виявляти аномалії, сповіщати про інциденти та швидко реагувати на загрози ще до того, як вони вплинуть на користувачів або бізнес-процеси.

До основних переваг моніторингу в реальному часі належать:

- Оперативність реагування. Завдяки мінімальній затримці між збором і аналізом даних, IT-команди можуть швидко реагувати на інциденти, мінімізуючи наслідки збоїв або атак.
- Проактивне управління інфраструктурою. Системи можуть автоматично надсилати сповіщення при відхиленні показників від норми, що дозволяє виявляти проблеми до того, як вони спричинять критичні збої.

- Моніторинг трендів. Довготривале спостереження за даними в реальному часі допомагає виявляти закономірності, прогнозувати навантаження та краще планувати розвиток інфраструктури.
- Покращення безпеки. Завдяки постійному аналізу журналів подій, мережевого трафіку та системного доступу можна швидко виявляти спроби несанкціонованого втручання чи потенційні вектори атак.
- Підвищення ефективності роботи персоналу. Надійна робота інфраструктури та відсутність затримок дозволяють співробітникам ефективно використовувати робочий час, не відволікаючись на технічні проблеми.
- Оптимізація витрат. Своєчасне виявлення та усунення несправностей знижує потребу в дорогому ремонті, модернізації або ліквідації наслідків інцидентів. Також можна уникнути простоїв, які впливають на дохід підприємства.[17]

Втім, варто також враховувати і деякі недоліки цього підходу:

- Надлишкові сповіщення. Автоматизовані системи іноді генерують багато другорядних або хибно позитивних повідомлень, що ускладнює виявлення дійсно критичних інцидентів без участі фахівця.
- Обмеження на зберігання даних. Висока швидкість генерування логів створює потребу в значних обсягах сховища. Часто доводиться видаляти або архівувати старі дані, що ускладнює довгостроковий аналіз.
- Проблеми масштабування. Більшість систем моніторингу працюють за моделлю SaaS, де ціна залежить від обсягу оброблюваних даних. Розширення бізнесу може призвести до значного зростання витрат.
- Складність інтерфейсу. Багато рішень мають перевантажений або неінтуїтивний інтерфейс, розрахований на досвідчених технічних користувачів. Це збільшує ризик помилок у роботі та ускладнює навчання нових співробітників.[18]

Отже, моніторинг у реальному часі — це не просто технологічне рішення, а стратегічний інструмент, що дозволяє забезпечити стабільну, безпечну та ефективну роботу інформаційних систем. За умови правильного впровадження він

допомагає не лише реагувати на інциденти, а й попереджати їх, зберігаючи безперервність і конкурентоспроможність бізнесу.

2.3 Огляд SIEM-систем та його надбудову

SIEM (системи управління інцидентами та подіями безпеки) – це рішення безпеки, яке допомагає організаціям розпізнавати й усунути потенційні загрози та вразливості безпеки до того, як у них з’явиться шанс порушити бізнес-операції.

Системи SIEM допомагають групам безпеки підприємства виявляти аномалії поведінки користувачів і використовувати штучний інтелект (AI) для автоматизації багатьох ручних процесів, пов’язаних із виявленням загроз і реагуванням на інциденти .

Початкові платформи SIEM були інструментами керування журналами. Вони поєднали функції керування інформацією про безпеку (SIM) і керування подіями безпеки (SEM). Ці платформи забезпечували моніторинг і аналіз подій, пов’язаних із безпекою, у реальному часі. Крім того, вони полегшили відстеження та реєстрацію даних безпеки для цілей відповідності або аудиту. Gartner ввів термін SIEM для поєднання технологій SIM і SEM у 2005 році.

Протягом багатьох років програмне забезпечення SIEM розвивалося, щоб включити аналітику поведінки користувачів і об’єктів (UEBA) , а також інші передові засоби аналітики безпеки, штучний інтелект і можливості машинного навчання для виявлення аномальної поведінки та індикаторів передових загроз. Сьогодні SIEM став основним елементом сучасних центрів безпеки (SOC) для моніторингу безпеки та управління відповідністю. Тож як працює SIEM?

На самому базовому рівні всі рішення SIEM виконують певний рівень функцій агрегації, консолідації та сортування даних для виявлення загроз і дотримання вимог щодо відповідності даних. Хоча деякі рішення відрізняються за можливостями, більшість пропонує однаковий базовий набір функцій:

- Керування журналами

SIEM отримує дані про події з широкого спектру джерел у всій IT-інфраструктурі організації, включаючи локальні та хмарні середовища. Дані журналу подій від користувачів, кінцевих точок, додатків, джерел даних, хмарних робочих навантажень і мереж, а також дані апаратного та програмного забезпечення безпеки, наприклад брандмауерів або антивірусного програмного забезпечення, збираються, співвідносяться та аналізуються в режимі реального часу.

Деякі рішення SIEM також інтегруються зі сторонніми каналами аналізу загроз , щоб співвіднести дані внутрішньої безпеки з попередньо розпізнаними сигнатурами та профілями загроз. Інтеграція з новинами про загрози в реальному часі дозволяє командам блокувати або виявляти нові типи сигнатур атак.

- Кореляція подій та аналітика

Кореляція подій є важливою частиною будь-якого рішення SIEM. Використовуючи розширену аналітику для виявлення та розуміння складних шаблонів даних, кореляція подій надає інформацію для швидкого визначення та пом’якшення потенційних загроз безпеці бізнесу.

Рішення SIEM значно покращують середній час виявлення (MTTD) і середній час відповіді (MTTR) для груп IT-безпеки, розвантажуючи ручні робочі процеси, пов’язані з поглибленим аналізом подій безпеки.

- Моніторинг інцидентів і сповіщення безпеки

SIEM консолідує свій аналіз в єдину центральну інформаційну панель, де групи безпеки відстежують діяльність, сортують сповіщення, ідентифікують загрози та ініціюють реагування або виправлення.

Більшість інформаційних панелей SIEM також включають візуалізацію даних у реальному часі, яка допомагає аналітикам безпеки виявляти сплески або тенденції підозрілої активності. Використовуючи настроювані попередньо визначені правила кореляції, адміністратори можуть негайно отримувати сповіщення та вживати відповідних заходів для пом’якшення загроз, перш ніж вони матеріалізуються у більш значні проблеми безпеки.

- Управління відповідністю та звітність

Рішення SIEM є популярним вибором для організацій, які підпадають під різні форми дотримання нормативних вимог. Завдяки автоматичному збору та аналізу даних, які він забезпечує, SIEM є цінним інструментом для збору та перевірки даних відповідності в усій бізнес-інфраструктурі.

Рішення SIEM можуть генерувати звіти про відповідність PCI-DSS, GDPR, HIPAA, SOX та інших стандартів відповідності в реальному часі, зменшуючи навантаження на управління безпекою та раннє виявлення можливих порушень, щоб їх можна було усунути. Багато рішень SIEM постачаються з готовими додатками, які можуть створювати автоматичні звіти, розроблені відповідно до вимог відповідності.

До або після того, як ви інвестували у своє нове рішення, ось деякі найкращі методи впровадження SIEM, яких вам слід дотримуватися:

1. Почніть із повного розуміння обсягу вашої реалізації. Визначте, як ваш бізнес отримає найкращу користь від розгортання, і налаштуйте відповідні варіанти використання безпеки.

2. Розробіть і застосуйте попередньо визначені правила кореляції даних у всіх системах і мережах, включаючи будь-які хмарні розгортання.

3. Визначте всі вимоги щодо відповідності вашого бізнесу та допоможіть переконатися, що ваше рішення SIEM налаштовано на аудит і звітування за цими стандартами в режимі реального часу, щоб ви могли краще зрозуміти свій ризик.

4. Каталогізуйте та класифікуйте всі цифрові активи в IT-інфраструктурі вашої організації . Це важливо під час керування збором даних журналу, виявлення зловживань доступом і моніторингу мережевої активності .

5. Встановіть політики BYOD , IT-конфігурації та обмеження, які можна контролювати під час інтеграції вашого рішення SIEM.

6. Регулярно налаштовуйте конфігурації SIEM, щоб зменшити кількість помилкових спрацьовувань у сповіщеннях безпеки.

7. Документуйте та відпрацьовуйте всі плани реагування на інциденти та робочі процеси, щоб гарантувати, що команди зможуть швидко реагувати на будь-які інциденти безпеки, які потребують втручання.

8. Автоматизуйте, де це можливо, за допомогою штучного інтелекту та технологій безпеки, таких як SOAR.

9. Оцініть можливість інвестування в керованого постачальника послуг безпеки (MSSP) для керування розгортанням SIEM. Залежно від унікальних потреб вашого бізнесу, MSSP можуть бути краще оснащені, щоб впоратися зі складнощами впровадження SIEM, а також регулярно керувати та підтримувати його безперервні функції. [8]

2.3.1 SOAR

SIEM системи особливо корисні для підприємств з великою інформаційною інфраструктурою, де важко відстежувати всі події. Вони надають спеціалістам з кібербезпеки чітку картину того, що відбувається в різних частинах інформаційної системи. Це полегшує роботу експертів і підвищує загальний рівень захисту IT інфраструктури. Перейшовши до другого аспекту еволюції кібербезпеки – SOAR (Security Orchestration Automation and Response). SOAR не лише доповнює можливості SIEM, але і розширює їх, надаючи більше автоматизації та інтеграції в процесах захисту і реагування на кіберзагрози. Розглянемо SOAR більш детально і розкриємо, як ця система сприяє вдосконаленню кібербезпеки організації.

Отже, SOAR – це набір важливих сервісів та інструментів, спрямованих на автоматизацію та оптимізацію процесів запобігання кібератакам та реагування на них. SOAR, як можна здогадатися зі скорочення, складається із трьох ключових компонентів, які спільно працюють для підвищення рівня кібербезпеки. [9]

Оркестровка безпеки (Security orchestration)

SOC використовують різні рішення для моніторингу та реагування на загрози, як-от брандмауери, канали розвідки про загрози та засоби захисту кінцевих точок. Навіть прості процеси безпеки можуть включати кілька інструментів. Наприклад, аналітику безпеки, який досліджує фішинговий електронний лист,

може знадобитися захищений шлюз електронної пошти, платформа аналізу загроз і антивірусне програмне забезпечення, щоб ідентифікувати, зрозуміти та усунути загрозу. Ці інструменти часто надходять від різних постачальників і можуть не легко інтегруватися, тому аналітики повинні вручну переміщатися між інструментами під час роботи.

За допомогою SOAR SOC можуть об'єднати ці інструменти в узгоджені, повторювані робочі процеси безпеки (SecOps). SOAR використовують інтерфейси прикладного програмування (API) , попередньо створені плагіни та спеціальні інтеграції для підключення інструментів безпеки (і деяких інструментів, не пов'язаних із безпекою). Після інтеграції цих інструментів SOC можуть координувати свою діяльність із посібниками.

Посібники – це карти процесів, які аналітики безпеки можуть використовувати для окреслення кроків стандартних процесів безпеки, таких як виявлення загроз, дослідження та реагування. Playbooks можуть охоплювати кілька інструментів і програм. Вони можуть бути повністю автоматизованими, повністю ручними або поєднувати автоматизовані та ручні завдання.

Автоматизація безпеки (Security automation)

Рішення безпеки SOAR можуть автоматизувати низькорівневі, трудомісткі, повторювані завдання, такі як відкриття та закриття запитів у службу підтримки, збагачення подій і визначення пріоритетів сповіщень. SOAR також можуть ініціювати автоматизовані дії інтегрованих інструментів безпеки. Це означає, що аналітики безпеки можуть використовувати робочі процеси, щоб об'єднати кілька інструментів і виконувати більш складну автоматизацію операцій безпеки.

Наприклад, розглянемо, як платформа SOAR може автоматизувати розслідування скомпрометованого ноутбука. Перша ознака того, що щось не так, надходить від рішення для виявлення та реагування кінцевих точок (EDR) , яке виявляє підозрілу активність на ноутбуці. EDR надсилає сповіщення до SOAR, який ініціює сценарій для виконання попередньо визначеної ігри. По-перше, SOAR відкриває заяву про інцидент. Він збагачує сповіщення даними з інтегрованих каналів аналізу загроз та інших інструментів безпеки. Потім SOAR виконує

автоматичні відповіді, такі як запуск інструменту виявлення та реагування мережі (NDR), щоб помістити кінцеву точку в карантин або спонукає антивірусне програмне забезпечення знайти та усунути шкідливе програмне забезпечення. Нарешті, SOAR передає квиток аналітику безпеки, який визначає, чи вирішено інцидент чи потрібне втручання людини.

Деякі SOAR включають штучний інтелект (AI) і машинне навчання, які аналізують дані з інструментів безпеки та рекомендують шляхи боротьби з загрозами в майбутньому.

Реагування на інцидент (Incident response)

Можливості оркестровки та автоматизації SOAR дозволяють йому служити центральною консоллю для реагування на інциденти безпеки (IR). У звіті IBM Cost of a Data Breach було виявлено, що організації, у яких є як команда IR, так і план тестування IR, виявили порушення на 54 дні швидше, ніж ті, у яких немає жодного.

Аналітики безпеки можуть використовувати SOAR для розслідування та вирішення інцидентів, не переходячи між кількома інструментами. Подібно до платформ аналізу загроз , SOAR збирає показники та сповіщення із зовнішніх каналів і інтегрованих інструментів безпеки на центральній інформаційній панелі. Аналітики можуть співвідносити дані з різних джерел, відфільтровувати хибні спрацьовування, визначати пріоритетність сповіщень і визначати конкретні загрози, з якими вони мають справу. Тоді аналітики можуть відповісти, запустивши відповідні ігри.

SOC також можуть використовувати інструменти SOAR для перевірок після інцидентів і більш проактивних процесів безпеки. Інформаційні панелі SOAR можуть допомогти командам безпеки зрозуміти, як певна загроза зламала мережу, і як запобігти подібним загрозам у майбутньому. Подібним чином групи безпеки можуть використовувати дані SOAR для виявлення непомічених поточних загроз і зосередити свої зусилля по пошуку загроз у потрібних місцях.Важливо відзначити, що SOAR рішення обмежуються функціональністю інтегрованих засобів кіберзахисту, якими можна віддалено керувати. Взаємодія із інтегрованими

продуктами здійснюється через API, і доступні можливості залежать від механізмів керування самого інтегрованого продукту. [10]

2.3.2 Порівняння найкращих SIEM-рішень на ринку для вашого бізнесу

У сучасному середовищі кіберзагроз системи управління інформаційною безпекою та подіями (SIEM) відіграють ключову роль у виявленні, аналізі та реагуванні на інциденти безпеки. Вони поєднують збір журналів, кореляцію подій, аналітику та автоматизацію реагування. З огляду на різноманіття рішень на ринку, нижче представлено порівняльну таблицю 2.2. провідних SIEM-інструментів, яка допоможе оцінити їх функціональні можливості.

Таблиця 2.2.Порівняльна таблиця провідних SIEM-інструментів

Назва інструменту	**Тип SIEM**	**Розгортання**	**Аналітика / AI**	**Підтримка SOAR**	**Основні функції**	**Ціна**
Log360 (ManageEngine)	Traditional	Локально / SaaS	Машинне навчання, UEBA	Часткова інтеграція	Аудит змін, збір логів, моніторинг AD	Платна, від $300/рік
Splunk Enterprise Security	Next-Gen	Локально / SaaS	Потужна кореляція, ML-моделі, візуалізація	Так (SOAR через Splunk Phantom)	Аналіз журналів, мультихмарна підтримка	Платна, ціна за запитом
IBM QRadar	Traditional / Hybrid	Локально / Хмара	Аналітика загроз, AI-аналітика Watson	Так (SOAR з QRadar SOAR)	Кореляція подій, інтеграція з IDS/IPS	Платна, ціна за запитом

Продовження таблиці 2.2

Назва інструменту	**Тип SIEM**	**Розгортання**	**Аналітика / AI**	**Підтримка SOAR**	**Основні функції**	**Ціна**
AT&T Cybersecurity (AlienVault USM)	Next-Gen	Локально / Хмара	Аналітика подій, класифікація загроз	Часткова інтеграція	SIEM + вразливості + аналіз загроз	Підписка, ціна за запитом
Exabeam Fusion	Next-Gen	Хмара	Поведінковий аналіз, автоматичні сценарії, UEBA	Так (SOAR інтегрований)	Повна історія інцидентів, детекція загроз	За запитом, підписка
Datadog Security Monitoring	Cloud-native SIEM	Хмара	Понад 400 інтеграцій, реальний час, ML	Часткова інтеграція (через Datadog workflows)	Cloud SIEM, попередні правила, сповіщення	Підписка, ціна за запитом
LogRhythm NextGen SIEM	Next-Gen	Локально / Хмара	AI-аналітика, UEBA, аналітика поведінки	Так (LogRhythm SOAR)	Швидке реагування, автоматизація інцидентів	Платна, ціна за запитом

Отже обидві системи допомагають підвищити рівень кібербезпеки IT-інфраструктури та компенсувати нестачу спеціалістів у сфері безпеки. Коли SIEM та SOAR системи працюють разом, вони доповнюють сильні та компенсують слабкі сторони одне одного. SIEM забезпечує централізований збір і аналіз даних, в той час як SOAR надає автоматизацію та координацію дій у разі інцидентів. Вони створюють більш комплексний підхід до кібербезпеки, допомагаючи організаціям ефективно захищати свою інформаційну інфраструктуру.

Сучасні SIEM та SOAR системи стають необхідними інструментами для забезпечення безпеки в динамічному світі кіберзагроз. Вони доповнюють одне одного і спільно створюють надійний захист від сучасних кібератак, допомагаючи організаціям бути кроком попереду потенційних загроз.

Висновки до розділу 2

У сучасних інформаційно-комунікаційних системах моніторинг логів відіграє ключову роль у забезпеченні безпеки. Основними етапами є агрегація, нормалізація, аналіз, зберігання та візуалізація логів. Ефективна робота з логами дозволяє своєчасно виявляти аномалії, інциденти та порушення політик безпеки.

SIEM-системи автоматизують процеси збору, обробки та кореляції подій безпеки з різних джерел, забезпечуючи централізований контроль, виявлення загроз і відповідність вимогам регуляторів.

SOAR-системи, у свою чергу, розширюють можливості SIEM за рахунок інтеграції інструментів, автоматизації рутинних завдань та керування інцидентами за наперед визначеними сценаріями (playbooks). Їх впровадження дозволяє зменшити час реакції на загрози, покращити ефективність аналітиків SOC та стандартизувати процеси реагування.

Комплексне використання SIEM і SOAR у поєднанні з найкращими практиками дає змогу створити надійну та адаптивну систему інформаційної безпеки.

Розділ 3 АНАЛІЗ ЛОГІВ ТА ВИЯВЛЕННЯ ІНЦИДЕНТІВ

Аналіз журналу – це процес перегляду та інтерпретації файлів журналу, щоб отримати уявлення про поведінку, продуктивність і безпеку системи. За допомогою процесу аналізу журналу ви можете визначити помилки, тенденції, шаблони, аномалії чи іншу важливу інформацію, яка може допомогти вам зрозуміти, як функціонує ваша система.

Журнали генеруються різними джерелами, такими як операційні системи, програми, бази даних, сервери, мережеві пристрої тощо. Кожне джерело має власний формат, важливість, структуру для реєстрації даних та рекомендований час зберігання (менеджмент логів наведено в Таблиці 3.1). Наприклад, журнали веб-сервера міститимуть інформацію про запити, зроблені до сервера, зокрема IP-адреси, Коди відповіді (Response codes), Агенти користувачів (User agents)

3.1 Методи аналізу логів

У сучасному світі більшість аналізу журналів виконується за допомогою інструментів і програмного забезпечення, ось деякі основні визначення використовуваних методів:

1. Кореляція

Кореляція є життєво важливою технікою аналізу журналів, яка передбачає пошук шаблонів або зв’язків між різними джерелами журналів. Наприклад, співвіднесення журналів доступу з журналами помилок може допомогти визначити будь-які помилки, які сталися під час сеансу певного користувача.

Ця техніка також відома як кореляція журналу і вона відіграє вирішальну роль у виявленні основної причини проблеми.

2. Аналіз продуктивності системи

Цей метод передбачає аналіз журналів, щоб отримати уявлення про продуктивність системи. Продуктивність можна виміряти за використанням ЦП, пам’яті та мережевим трафіком. Це виявляє вузькі місця системи, які необхідно усунути, а також покращує ефективність системи. Системні аналітики можуть використовувати певну форму аналізу журналів, щоб краще використовувати великі обсяги створених системних даних.

3. Аналіз журналів за допомогою ШІ

Аналіз журналів за допомогою штучного інтелекту – це техніка, яка поєднує штучний інтелект і машинне навчання для автоматизації аналізу журналів і сповіщення про будь-які значні події чи аномалії. Це покращує ефективність аналізу журналів за рахунок зменшення ручних зусиль і надання інформації в реальному часі.

4. Розпізнавання образів

Розпізнавання шаблонів – це техніка, яка передбачає аналіз шаблонів журналу для виявлення аномалій. Наприклад, ви можете використовувати розпізнавання шаблонів для виявлення незвичайних стрибків у трафіку або шаблонів повторюваних помилок. [19]

3.2 Кореляція логів

Кореляція логів — це процес збору та аналізу даних журналу з різних джерел для виявлення потенційних порушень безпеки. Це включає агрегування та нормалізацію логів з різних систем, таких як антивірусне програмне забезпечення, брандмауери та мережеві пристрої, щоб створити повне уявлення про мережеву активність. Мета полягає в тому, щоб виявити шаблони та аномалії, які можуть вказувати на загрозу безпеці.

Кореляція журналів або кореляція файлів журналів є важливою практикою, яка передбачає процес збору даних із кількох джерел, їх оцінки та визначення того, як різні набори даних можуть бути пов’язані між собою. Це важлива функція систем управління інформацією про безпеку та подіями (SIEM), які відстежують і аналізують сповіщення безпеки, створені в комп’ютерному середовищі.

Діяльність у сфері кібербезпеки генерує величезні обсяги логів – від антивірусного програмного забезпечення до систем виявлення вторгнень, брандмауерів і навіть самих операційних систем. Кожна дія, помилка та попередження створює файл логу. Окремо ці журнали надають дуже конкретну інформацію про поведінку програми, показуючи все: від невдалих спроб входу до виявлених загроз.

Кореляція логів складається з двох основних етапів: збору та аналізу. Першим кроком є збір усіх даних журналу та створення центрального сховища цієї інформації. Організація цих даних у систематичний спосіб є важливою, враховуючи обсяг і потенційну складність задіяних наборів даних. Виникає необхідність об’єднати всі дані безпеки в одну мову, щоб спростити аналіз і кореляцію.

Другим кроком є аналіз цих логів, щоб виявити будь-які потенційні зв’язки чи кореляції. Програмне забезпечення загалом спрощує процес кореляції завдяки системі, призначеній для позначення певних послідовностей або порушень, які можуть свідчити про загрозу безпеці. Автоматизована кореляція журналів може допомогти командам із кібербезпеки швидко виявляти потенційні загрози та скорочувати час між початковою компрометацією та пом’якшенням.

У контексті антивірусної програми кореляція логів може допомогти ідентифікувати раніше невідоме шкідливе програмне забезпечення або сигнатури вірусів. Наприклад, антивірусне програмне забезпечення могло виявити невідому загрозу та зареєструвати її. Пізніше кореляція може виявити, що та сама загроза також з’явилася деінде в мережі. Аналіз цієї корелюючої поведінки може призвести до визначення нової сигнатури вірусу, що дасть змогу знищити цю щойно виявлену загрозу.

Основна перевага кореляції логів у сфері кібербезпеки стає належним чином очевидним, коли в мережі відбувається значна діяльність поза межами місця. Завдяки кореляції журналів виявляються незначні зміни в кількох системах, які зазвичай можуть залишатися непоміченими, сповіщаючи команди безпеки про можливі порушення чи інші проблеми.

У той же час кореляція логів також може надати історичний контекст для порушень, надаючи командам необхідну перспективу того, як порушення могло статися. Він може підтримати розслідування витоку та запропонувати хронологічно точне зображення або «хронологію» подій, що призвели до порушення.

Зловмисники часто залишають сліди або «сліди», зазвичай у формі незнайомих записів або дивної поведінки, записаної в логах. Спостереження за цими слідами та порівняння їх із звичайною поведінкою допомагає виявити підозрілу діяльність. Ось тут і вступає в дію сила кореляції — можливість висвітлити подібні аномалії в різних системах цілком може стати першим попередженням про організовану кібератаку.

Оперативна інформація з кореляції логів може бути важливою для оптимального розподілу ресурсів. Розуміючи повторювані шаблони системних або мережевих проблем, команди можуть зосередити ресурси на усуненні повторюваних проблем, а не проводити спеціальне гасіння пожеж. Це не тільки покращує функції систем, але й дозволяє краще розподіляти ресурси безпеки.

Незважаючи на те, що це здається простим і зрозумілим процесом, кореляція логів може стати складною через безпрецедентну кількість даних, що генеруються щодня. Важливо відзначити, що ефективне використання відображення зібраних і корельованих даних може змінити спосіб виявлення та пом’якшення загроз кібербезпеці. [13]

Кореляція журналів є незамінним інструментом у кібербезпеці та антивірусних операціях. Вона має потенціал для швидкого виявлення потенційних загроз, забезпечуючи негайне пом’якшення, спрощення операцій і забезпечення безперебійної роботи мережевих середовищ, забезпечуючи систему раннього попередження, яка завчасно визначає загрози безпеці. Включення кореляції

журналів у системи безпеки сприяє створенню безпечнішого та надійнішого мережевого середовища.

3.3 Використання штучного інтелекту та машинного навчання для аналізу логів

Організації стають дедалі краще самостійно виявляти витоки даних, 53 відсотки зламів було виявлено зовнішнім джерелом у 2017 році тобто організації не підозрювали, що їхні дані зламані. Частково проблема полягає в тому, що для багатьох організацій немає простого способу автоматично співвідносити та аналізувати всі дані, зібрані різними рішеннями безпеки, які були розгорнуті в мережі. Ця проблема ускладнюється тим, що багато з цих інструментів працюють ізольовано. У результаті IT-командам доводиться передавати корелятивні дані, зібрані з різних джерел, шукаючи голку в стозі сіна. Можливість людської помилки висока, і файли журналів просто прокручуються занадто швидко, щоб будь-хто міг зібрати з них корисну інформацію.

Саме тому багато організацій — як кінцеві користувачі, так і постачальники — звертаються до машинного навчання, щоб заповнити прогалину. Машинне навчання — це галузь штучного інтелекту, яка використовує алгоритми, які дозволяють системам точніше прогнозувати результати шляхом аналізу даних, визначення закономірностей і прийняття рішень із мінімальним втручанням людини. Цей метод використовує швидкість, ефективність і точність технології, щоб не тільки знаходити моделі поведінки та ознаки компрометації, які люди інакше б пропустили, але й реагувати на ці загрози майже в реальному часі. [6]

Машинне навчання (ML) — це галузь штучного інтелекту (ШІ), яка зосереджена на тому, щоб дозволити комп’ютерам і машинам імітувати спосіб навчання людей, виконувати завдання автономно та покращувати свою

продуктивність і точність за допомогою досвіду та доступу до більшої кількості даних. У цьому контексті існує три основні підходи:

Контрольоване навчання, визначається використанням позначених наборів даних для навчання алгоритмів для класифікації даних або точного прогнозування результатів. Коли вхідні дані надходять у модель, модель коригує свої ваги, поки не буде встановлено належним чином. Це відбувається як частина процесу перехресної перевірки, щоб переконатися, що модель не переобладнана або не підігнана. Контрольоване навчання допомагає організаціям вирішувати різноманітні реальні проблеми в масштабі, наприклад, класифікувати спам в окремій папці з папки "Вхідні". Деякі методи, які використовуються в керованому навчанні, включають нейронні мережі, наївну байєсовську регресію, лінійну регресію, логістичну регресію, випадковий ліс і машину опорних векторів (SVM).

Неконтрольоване навчання, використовує алгоритми машинного навчання для аналізу та кластеризації непозначених наборів даних (підмножин, які називаються кластерами). Ці алгоритми виявляють приховані шаблони або групи даних, без втручання людини. Здатність неконтрольованого навчання виявляти схожість і відмінності в інформації робить його ідеальним для дослідницького аналізу даних, стратегій перехресних продажів, сегментації клієнтів і розпізнавання зображень і шаблонів. Він також використовується для зменшення кількості функцій у моделі через процес зменшення розмірності. Аналіз головних компонентів (PCA) і сингулярне розкладання (SVD) є двома поширеними підходами для цього. Інші алгоритми, які використовуються в неконтрольованому навчанні, включають нейронні мережі, кластеризацію k-середніх і ймовірнісні методи кластеризації.

Напівконтрольоване навчання є посередником між контрольованим і неконтрольованим навчанням. Під час навчання він використовує менший набір даних з мітками, щоб керувати класифікацією та виділенням ознак із більшого набору даних без міток. Напівконтрольоване навчання може вирішити проблему відсутності достатньої кількості позначених даних для алгоритму контрольованого

навчання. Це також допомагає, якщо надто дорого позначити достатню кількість даних.

Навчання з підкріпленням – це модель машинного навчання, схожа на контрольоване навчання, але алгоритм не навчається за допомогою вибіркових даних. Ця модель навчається по ходу за допомогою проб і помилок. Послідовність успішних результатів буде посилена, щоб розробити найкращу рекомендацію чи політику щодо певної проблеми.

Зазвичай використовується великий ряд алгоритмів машинного навчання. До основних та більш дієвих них належать:

- Нейронні мережі імітують роботу людського мозку за допомогою величезної кількості зв'язаних вузлів обробки. Нейронні мережі добре розпізнають шаблони та відіграють важливу роль у програмах, включаючи переклад природної мови, розпізнавання зображень, розпізнавання мовлення та створення зображень.
- Лінійна регресія цей алгоритм використовується для прогнозування числових значень на основі лінійної залежності між різними значеннями. Наприклад, цю техніку можна використовувати для прогнозування цін на житло на основі історичних даних для регіону.
- Логістична регресія цей контрольований алгоритм навчання робить прогнози для категоричних змінних відповідей, таких як відповіді «так/ні» на запитання. Його можна використовувати для таких програм, як класифікація спаму та контроль якості на виробничій лінії.
- Кластеризація використовуючи неконтрольоване навчання, алгоритми кластеризації можуть ідентифікувати шаблони в даних, щоб їх можна було згрупувати. Комп'ютери можуть допомогти дослідникам даних, визначаючи відмінності між елементами даних, які люди не помітили.
- Дерева рішень можна використовувати як для прогнозування числових значень (регресія), так і для класифікації даних за категоріями. Дерева рішень використовують розгалужену послідовність пов'язаних рішень, які можна представити за допомогою деревоподібної діаграми. Однією з переваг дерев рішень

є те, що їх легко перевірити та перевірити, на відміну від чорного ящика нейронної мережі.

- У випадковому лісі алгоритм машинного навчання передбачає значення або категорію, поєднуючи результати з кількох дерев рішень.[15]

Глибоке навчання — це потужна форма машинного навчання, яка працює шляхом навчання нейронних мереж на величезних обсягах даних для пошуку закономірностей. Цей підхід часто використовується з контрольованим навчанням із позначеними наборами даних.

Дослідження під назвою «Deeplog» з Університету Юти демонструє один приклад використання глибокого навчання для журналів. У статті глибоке навчання використовується для виявлення аномалій у журналах і аналізу їх на типи подій, що може значно підвищити точність виявлення аномалій у журналах.

Проблема глибокого навчання полягає в тому, що йому потрібні великі обсяги даних, щоб стати точними, а це означає, що новим середовищам може знадобитися більше часу, щоб забезпечити точні прогнози.

Поглиблене навчання також може бути інтенсивним для обчислень і вимагати дорогих екземплярів графічного процесора для швидшого навчання моделей. З часом цей підхід може бути дорогим для виконання автоматизованого аналізу журналу.

Хоча організації використовують різні способи автоматизації аналізу журналів, вибір між згаданими вище методами стає вирішальним. Баланс між точністю та застосовністю в реальному світі може бути делікатним, тому важливо розуміти, як працюють ці методи. Інтеграція ШІ та машинного навчання в аналіз журналів має багато переваг:

- Швидше сортування даних: штучний інтелект може допомогти ефективно згрупувати схожі журнали та спростити пошук даних і навігацію.
- Легке виявлення проблем: штучний інтелект і машинне навчання чудово допомагають автоматизувати ідентифікацію проблем навіть із величезним обсягом журналів.

- Раннє виявлення аномалій: машинне навчання може допомогти запобігти інцидентам, вирішуючи проблеми на їх початку.
- Оптимізований розподіл ресурсів: AI/ML забезпечує швидкий і точний розподіл ресурсів для високопріоритетних областей, витрачаючи менше часу на ручний аналіз журналів.
- Сповіщення про важливу інформацію: використання штучного інтелекту та машинного навчання в аналізі журналів може допомогти звести до мінімуму помилкові сповіщення та допомогти організаціям переконатися, що сповіщення запускаються лише у випадках високого пріоритету.
- Аналіз журналів за допомогою AI та ML може допомогти компаніям підвищити ефективність керування даними журналів. Наявність правильного інструменту для реалізації ML з аналізом журналів є надзвичайно важливою. [15]

Отже еволюція аналізу журналів від традиційних процесів до об'єднання штучного інтелекту та методів машинного навчання свідчить про значний стрибок у технології та аналітиці даних. Ця еволюція змінила те, як групи обробки даних виявляють шаблони та аномалії в даних журналу, щоб вони могли витрачати менше часу на аналіз і зосередитися на більш важливих питаннях у своїй організації, наприклад на розробці продукту.

3.4 Розпізнавання образів

Розпізнавання образів — це підгалузь штучного інтелекту, яка передбачає ідентифікацію шаблонів і тенденцій на основі даних. Він дозволяє автоматично класифікувати дані на основі попередньо встановлених правил і визначень. У контексті кібербезпеки та антивірусів розпізнавання шаблонів має вирішальне значення для виявлення та запобігання кібератакам і зловмисному програмному забезпеченню. Віруси та інше шкідливе програмне забезпечення часто мають певні шаблони або характеристики, які дозволяють їх ідентифікувати та класифікувати.

За допомогою виявлення шаблонів і тенденцій експерти з безпеки та антивірусні програми можуть швидко виявляти та пом’якшувати загрози.

Технологія розпізнавання зображень — це область розпізнавання шаблонів, що застосовується в індустрії кібербезпеки та антивірусних програм на основі алгоритмів машинного навчання , які дозволяють аналізувати поведінку всіх мережевих подій. Розпізнавання зловмисної поведінки шляхом визначення певного шаблону, застосованого до більшості фрагментів коду, прикладом може слугувати (Рис.3.1) Оскільки інші фактори можуть спровокувати потенційне проникнення вірусу, як-от відвідані веб-сайти, це може викликати життєво важливі сигнали тривоги, які попереджають користувача про потенційні ризики та рекомендують не відвідувати сторінку. Простіше кажучи, він базується на передових алгоритмах і аналітичних методах, розроблених для розпізнавання шаблонів даних. Алгоритми розпізнавання шаблонів ідентифікують і аналізують шаблони даних з мільйонів файлів, поведінки, кодів, створюючи унікальні маркери, які використовуються для розпізнавання й уникнення порушень даних і вдосконалення протоколів безпеки джерел даних.

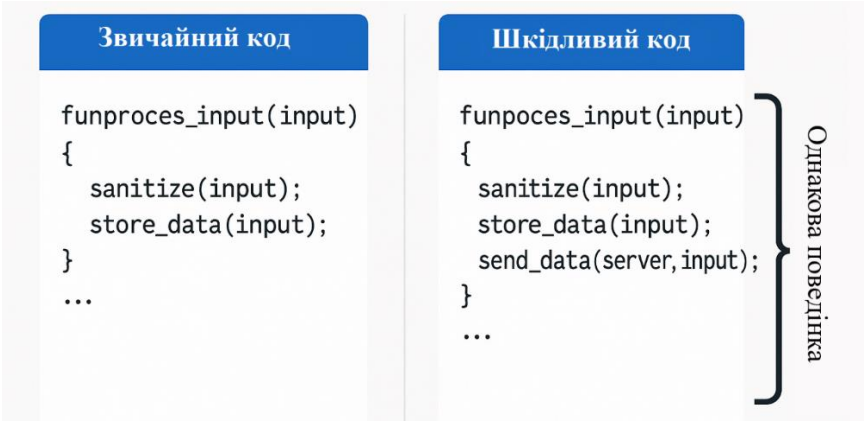


Рис. 3.1. Порівняння двох частин коду

Програмне забезпечення для розпізнавання образів працює з різними категоріями. Спочатку використовувався групами аналізу зловмисного програмного забезпечення, він аналізує та класифікує різні сімейства зловмисного програмного забезпечення за різними категоріями, які окреслюють найкращі

практики безпеки. Коли виявлено нові сімейства, це стає інформативною категорією для власників, IT-персоналу та типових засобів безпеки (брандмауери, проксі-сервери, IDS, IPS, кінцева точка, хмара, керування пристроями).

Сьогодні програмне забезпечення передового рівня пропонує візуальні інструменти, які аналізують кожен аспект шкідливого програмного забезпечення. Вони аналізують політики макросів, ключі реєстру, загальні файли або DLL програмного забезпечення. Ці нові аналітичні технології кібербезпеки виконують глибоке сканування, щоб класифікувати будь-які ризики або походження, пов'язані з: по суті, поділяють атаки на ті, які мають на меті пошкодити комп'ютер. Костюми та сценарії категорій виявлення зловмисного програмного забезпечення розгалужують стратегів аналізу, що значно вдосконалюють, і спеціалістів із впровадження кібербезпеки.

Має швидкий доступ до зібраної інформації, тому сучасні обмеження за обсягом і часом є основними для стандартного аналізу зловмисного програмного забезпеченняі виробництво антивірусних операцій. Деякі організації включають шаблони даних, поширеність загальноприйнятих переконань, можливо, зменшують частку фактів. Інші зберігають історичні показники, що спрощує ефективність аналізу кількох множинних розбіжностей подій, які відбулися в їхніх промислових успадкованих вертикалях. Більшість постачальників антивірусних програм обробляють дані, використовуючи алгоритми машинного навчання, алгоритми, спеціально застосовані для передбачення можливих ризиків, виявлення підвищених загроз, на які комп'ютер може бути виявлений, створюють автоматичні відповіді, зменшуючи робоче навантаження технічних захисників, пов'язаних з аналізом тонн стиснених даних, виконання відповідних подій відповідей на вторгнення в дані.

За останні роки експерти з кібербезпеки та постачальники антивірусних програм змогли успішно використовувати можливості розпізнавання шаблонів і машинного навчання. Оскільки сьогодні щодня відбувається обмін даними від мільйонів до мільярдів, системи передбачуваного аналізу тепер потребують додаткових додаткових прогнозних змінних, які завершать огляд рішень цифрової

безпеки . Хоча виявлення аномалій пропонує фактор ймовірності, що спонукає до спеціалізованого поведінкового аналізу, що призводить до одночасного проведення десятків глибоких перевірок, які завершуються на середньому пороговому значенні виявлення продуктивності при обробці більше ста тисяч — часто експоненціально натискаючи на ці сценарії моделювання, адаптовані до жорстких обмежень виробництва, створюючи можливі ризики як відмовостійкі.

Алгоритми розпізнавання образів не є надійними, і передбачуваний запобіжний механізм все ще є прагненням, оскільки вони не можуть повністю розрізнити, чи набір даних/уразливість походить від події в реальному часі чи розвідувальних операцій. Багато нещодавніх лейтмотивів щодо кіберзлочинності стосуються конфіденційних атак, застосованих за допомогою алгоритму фішингу, які також призводять до небажаних результатів, а також створюють проблеми через часті адаптації типів і концепцій фішингу (наприклад, компрометація бізнес-електронної пошти, китобійна підробка, зловживання ланцюгом поставок). [20]

Успіх розпізнавання образів залежить від ефективного аналізу даних. Машинні алгоритми та алгоритми метаданих приймають рішення, які виробляють розумні дії, а не просто порівнюють набори даних назад і вперед. Алгоритми машинного навчання надають можливість видобувати величезні обсяги структурованих і неструктурованих даних порівняно з людьми з точки зору часу дії, що дозволяє обійти персонал від роботи без стеження, пов’язаної з дотриманням політики, прямолінійним провалом формування кампаній скринінгу ефективності, пов’язаних із методологіями управління захистом від атак і загальним більш точним плануванням аналізу витрат на реакцію. Інтелектуальний аналіз даних дозволяє швидше та ефективніше приймати рішення завдяки швидкій обробці великих обсягів даних.

3.5 Виявлення ефективності вашої AI або ML моделі

Отже, Ви створили свою модель машинного навчання, та потрібно оцінити та підтвердити, наскільки це добре (чи погано) працює, щоб ви могли вирішити, чи варто це впроваджувати. В уьому допоможе крива AUC-ROC

Крива ROC, або крива робочих характеристик приймача, схожа на графік, який показує, наскільки добре працює модель класифікації. Це допомагає нам побачити, як модель приймає рішення на різних рівнях визначеності. Крива має дві лінії: одна показує, як часто модель правильно визначає позитивні випадки (TPR), а інша показує, як часто вона помилково визначає негативні випадки як позитивні (FPR). Дивлячись на цей графік, ми можемо зрозуміти, наскільки хороша модель, і вибрати поріг, який дає нам правильний баланс між правильними та неправильними прогнозами.

3.5.1 Receiver Operator Characteristic

Крива Receiver Operator Characteristic (ROC) — це метрика оцінки для задач двійкової класифікації. Це крива ймовірності, яка відкладає TPR проти FPR при різних порогових значеннях і по суті відокремлює «сигнал» від «шуму». Іншими словами, він показує продуктивність моделі класифікації за всіма пороговими значеннями класифікації. Площа під кривою (AUC) є мірою здатності двійкового класифікатора розрізняти класи та використовується як зведення кривої ROC.

Коли AUC = 1, як показано на (Рис.3.2), то класифікатор може правильно розрізнити всі Позитивні та Негативні точки класу. Проте, якби AUC був 0, тоді класифікатор передбачав би всі негативи як позитивні, а всі позитивні як негативні.

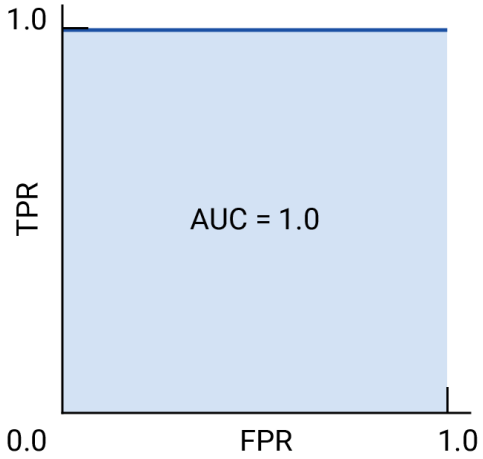


Рис. 3.2. ROC-крива з AUC = 1 (ідеальний класифікатор)[25]

Коли 0.5<AUC<1, як показано на (Рис.3.3), то існує висока ймовірність того, що класифікатор зможе відрізнити позитивні значення класу від негативних. Це відбувається тому, що класифікатор здатний виявляти більше істинних позитивних і справжніх негативних результатів, ніж помилкових негативних і помилкових позитивних результатів.

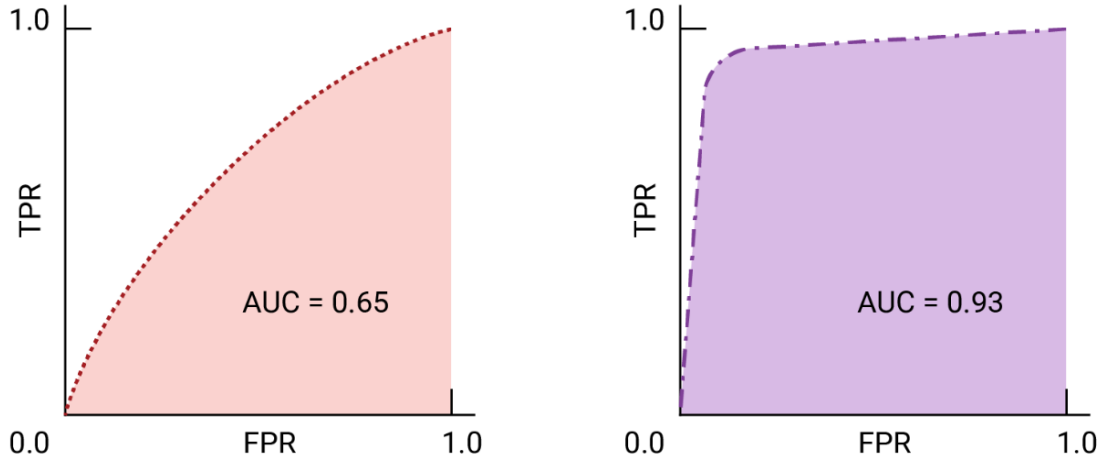


Рис. 3.3. ROC-крива з AUC між 0.5 і 1 (ефективний, але не ідеальний класифікатор).[25]

Коли AUC=0,5, як показано на (Рис.3.4) класифікатор не може розрізняти позитивні та негативні бали класу. Це означає, що класифікатор передбачає або випадковий клас, або постійний клас для всіх точок даних.

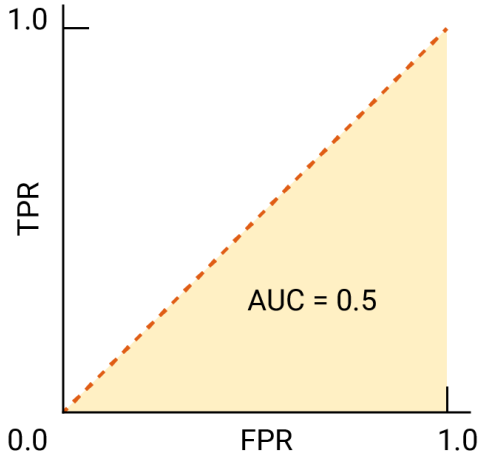


Рис. 3.4. ROC-крива з AUC = 0.5 (випадковий класифікатор)[25]

Крива ROC (робоча характеристика приймача) схожа на графік, який показує, наскільки добре модель розрізняє речі. Це допомагає нам побачити, як часто модель правильно визначає позитивні речі та як часто вона правильно уникає позначення негативних речей як позитивних. Отже, можна зробити висновок що, чим вище значення AUC для класифікатора, тим краще його ab, але не завжди так крива AUC-ROC також залежить від чутливісті і специфічністі.[23]

3.5.2 Чутливість і специфічність

Матриця плутанини – це інструмент візуалізації, який допомагає зрозуміти ефективність моделі класифікації. Це таблиця, яка підсумовує прогнози, зроблені моделлю, порівняно з фактичними значеннями істини (Рис.3.5).

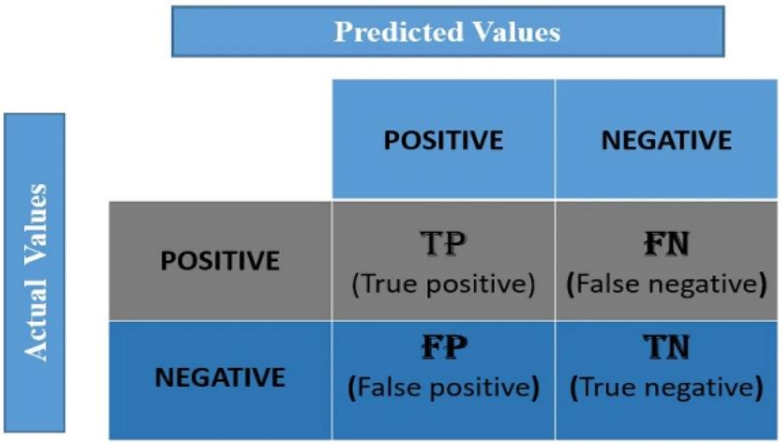


Рис. 3.5. Матриця плутанини

Є 4 терміни, які ви повинні розуміти, щоб правильно інтерпретувати або читати матрицю плутанини:

- Справжній позитивний (TP)
- Хибнопозитивний (FP)
- Справжній негатив (TN)
- Помилково негативний (FN)[24]

З матриці плутанини ми можемо вивести деякі важливі показники, які не обговорювалися в попередній статті. Давайте поговоримо про них тут.

1. Чутливість / Істинний позитивний коефіцієнт / Відкликання

Чутливість говорить нам, яка частка позитивного класу була правильно класифікована та обчислюється формулою 3.1.

Формула 3.1.

$$Sensitivity = \frac{TP}{TP + FN}$$

Простим прикладом може бути визначення того, яка частка фактично хворих людей була правильно виявлена моделлю.

2. Помилково негативний рейтинг

Частота помилкових негативних результатів (FNR) повідомляє нам, яку частку позитивного класу було неправильно класифіковано класифікатором та обчислюється формулою 3.2.

Формула 3.2.

$$FNR = \frac{FN}{TP + FN}$$

Вищий TPR і нижчий FNR є бажаними, оскільки ми хочемо правильно класифікувати позитивний клас.

3. Специфічність / Справжній негативний показник

Конкретність говорить нам, яку частку негативного класу було правильно класифіковано та обчислюється формулою 3.3.

Формула 3.3.

$$Specificity = \frac{TN}{TN + FP}$$

Беручи той самий приклад, що й у Чутливості, Специфічність означатиме визначення частки здорових людей, які були правильно ідентифіковані моделлю.

4. Хибнопозитивний рівень

FPR повідомляє нам, яка частка негативного класу була неправильно класифікована класифікатором та обчислюється формулою 3.4.

Формула 3.4.

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity$$

Вищий TNR і нижчий FPR є бажаними, оскільки ми хочемо правильно класифікувати негативний клас.

З цих показників Чутливість і Специфічність є, найважливішими, бо вони використовуються для побудови оціночної метрики.

3.5.3 Як працює крива AUC-ROC?

На кривій AUC-ROC (Рис.3.6) значення на осі X вказує на більшу кількість помилкових позитивних результатів, ніж справжніх негативних. Тоді як значення осі Y вказує на більшу кількість істинних позитивних результатів, ніж помилкових

негативних. Отже, вибір порогу залежить від здатності природним чином збалансувати помилкові позитивні та помилкові негативні результати.

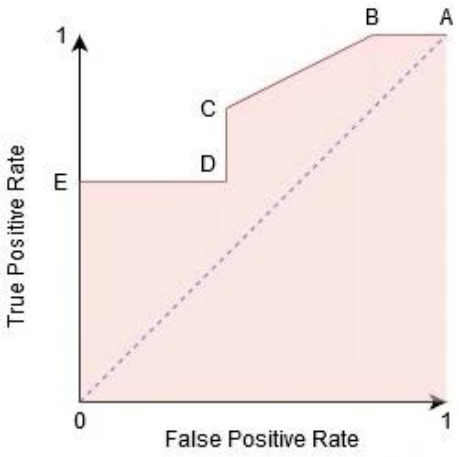


Рис. 3.6. Приклад практичної кривої AUC-ROC

Можемо спробувати зрозуміти цей графік, згенерувавши матрицю плутанини для кожної точки, що відповідає порогу, і поговорити про продуктивність нашого класифікатора.

Точка A – це місце, де чутливість найвища, а специфічність найнижча. Це означає, що всі позитивні бали класу класифікуються правильно, а всі негативні бали класифікуються неправильно. Відповідна матриця плутанини для цієї точки представлена на (Рис.3.7).

		Actual	
		+ve	-ve
Predicted	+ve	5	5
	-ve	0	0

Рис. 3.7. Матриця до точки A

Насправді будь-яка точка на синій лінії відповідає ситуації, коли коефіцієнт істинних позитивних результатів дорівнює коефіцієнту помилкових позитивних результатів.

Усі точки над цією лінією відповідають ситуації, коли частка правильно класифікованих точок, що належать до позитивного класу, більша, ніж частка неправильно класифікованих точок, що належать до негативного класу.

Хоча точка B має таку саму чутливість, як і точка A, вона має вищу специфічність. Це означає, що кількість неправильно негативних балів класу нижча за попередній поріг. Це означає, що цей поріг кращий за попередній. Відповідна матриця плутанини для цієї точки представлена на (Рис.3.8).

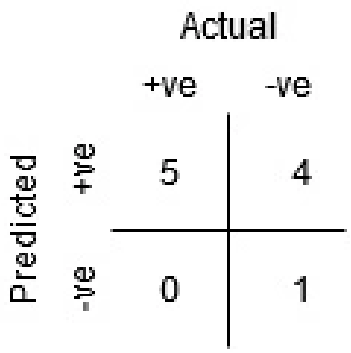


Рис. 3.8. Матриця до точки B

Між точками C і D чутливість у точці C вища, ніж у точці D для тієї самої специфічності. Це означає, що для такої ж кількості неправильно класифікованих негативних балів класу класифікатор передбачив більшу кількість позитивних балів класу. Тому поріг у точці C кращий, ніж у точці D. Відповідні матриці плутанини для цих точки представлені на (Рис.3.9).

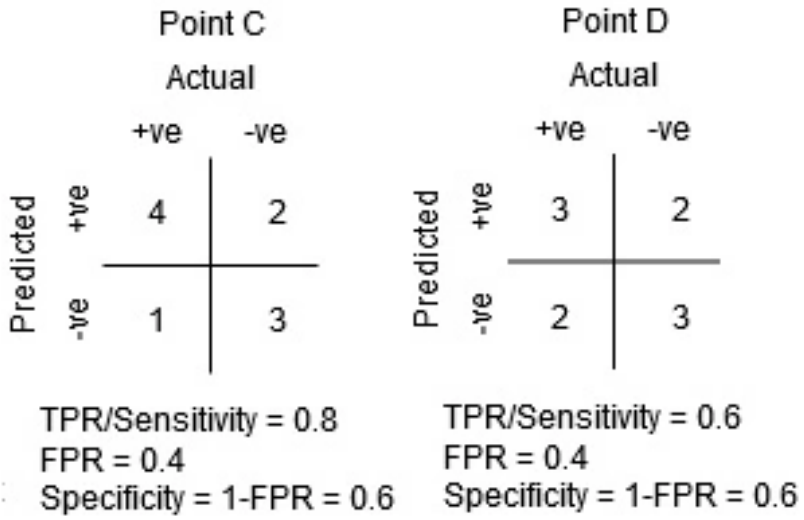


Рис. 3.9. Матриці до точок C і D

Точка E – це місце, де специфічність стає найвищою. Це означає, що модель не класифікує помилкові спрацьовування. Модель може правильно класифікувати всі бали негативного класу. Ми б вибрали цей пункт, якби нашою проблемою було надати ідеальні рекомендації щодо пісень нашим користувачам. Відповідна матриця плутанини для цієї точки представлена на (Рис.3.10). [23]

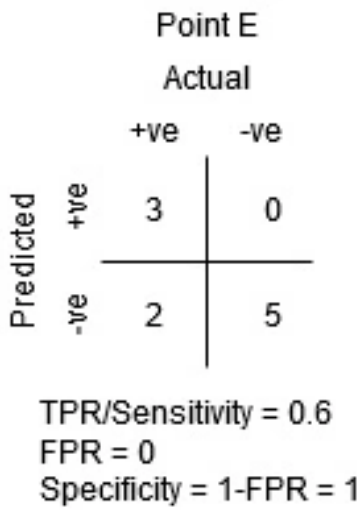


Рис. 3.10. Матриця до точки E

Отже оцінка ефективності моделей машинного навчання за допомогою кривої AUC-ROC та таких метрик, як чутливість і специфічність дозволяє глибше зрозуміти баланс між істинно позитивними та хибнопозитивними передбаченнями

на різних порогах прийняття рішень. Площа під кривою (AUC) є важливою характеристикою, що відображає здатність моделі розрізняти між класами, чим вищий показник AUC, тим краща загальна якість моделі. Разом із аналізом матриці плутанини і виведенням показників чутливості та специфічності, AUC-ROC забезпечує комплексне уявлення про продуктивність моделі. Правильний вибір порогу класифікації на основі цих метрик дозволяє адаптувати модель під конкретні завдання та досягати оптимального балансу між різними типами помилок.

Висновки до розділу 3

Ефективне управління логами є ключовим елементом забезпечення кібербезпеки сучасних інформаційних систем. У цьому розділі було розглянуто основні етапи аналізу логів: збір, зберігання, кореляція, аналіз та зберігання історії подій. Кожен із цих процесів має критичне значення для виявлення та реагування на потенційні інциденти безпеки. Кореляція подій допомагає встановити причинно-наслідкові зв’язки, що значно підвищує ефективність реагування. Також висвітлено роль штучного інтелекту (ШІ) та машинного навчання (ML) у вдосконаленні процесів аналізу логів. Методи ML — включаючи контрольоване, неконтрольоване, напівконтрольоване навчання та навчання з підкріпленням — дозволяють автоматизувати виявлення аномалій, передбачати загрози та підвищувати точність аналізу. Використання глибокого навчання, а також технологій розпізнавання образів та шаблонів, відкриває нові можливості для раннього виявлення складних кіберзагроз та зниження кількості помилкових спрацювань.

Крім того, важливим етапом використання методів машинного навчання є оцінка ефективності побудованих моделей. Для цього застосовуються метрики, такі як чутливість, специфічність, а також крива AUC-ROC, що дозволяють визначити здатність моделі правильно класифікувати події безпеки. Аналіз цих показників допомагає оптимізувати параметри моделі та забезпечити баланс між виявленням

загроз і мінімізацією кількості помилкових спрацювань, що критично важливо для побудови надійної системи кіберзахисту.

Таким чином, поєднання традиційних методів обробки логів з можливостями штучного інтелекту створює потужну основу для побудови проактивної системи кіберзахисту, здатної не лише вчасно виявляти інциденти, а й ефективно на них реагувати, знижуючи ризики для інформаційної безпеки організації.

Розділ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ АНАЛІЗУ ЛОГІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

Метою практичної частини дипломної роботи є побудова моделі контрольованого машинного навчання для автоматичного виявлення аномального мережевого трафіку на основі попередньо зібраних та нормалізованих логів.

Завдання полягає в реалізації повного циклу побудови моделі: від обробки вхідних даних до оцінки її ефективності при класифікації мережевих сесій на нормальні та потенційно небезпечні (атаки).

В результаті повинна бути створена і протестована модель, здатна:

- автоматично обробляти дані логів,
- навчатись на розмічених прикладах трафіку,
- здійснювати класифікацію з високою точністю,
- використовуватись у системах моніторингу інформаційної безпеки (наприклад, у зв’язці з SIEM/SOAR рішеннями).

Для побудови моделі машинного навчання були використані структуровані дані мережевого трафіку у вигляді CSV-файлів. Дані були отримані з папки TrafficLabelling, яка містить серію логів, попередньо оброблених та нормалізованих для подальшого аналізу.

Кожен файл містить такі основні поля:

- Flow ID – ідентифікатор сесії (унікальний),
- Source IP / Port – адреса та порт джерела,
- Destination IP / Port – адреса та порт призначення,
- Timestamp – час початку сесії,
- Label – цільова змінна, що містить розмітку (наприклад, "BENIGN", "DoS", "PortScan" тощо),
- Технічні характеристики сесій – середня швидкість, кількість байтів/пакетів, час затримки, частота передач тощо.

Загальна кількість оброблених логів склала N файлів. Було об’єднано всі логи, які містили стовпець Label, що є обов’язковим для навчання моделі класифікації.

4.1 Етапи обробки даних

Ці дані вже пройшли стадію первинної фільтрації та були адаптовані до формату, зручного для застосування алгоритмів машинного навчання.

Перед подачею в модель дані пройшли декілька етапів обробки:

1. Об’єднання вхідних CSV-файлів

Було автоматично зчитано всі CSV-файли з вказаної директорії. Лише ті файли, що містили колонку Label, були включені до навчального набору.

2. Очищення від неінформативних полів

Було видалено поля, які не несуть значного смислового навантаження або можуть спричинити перенавчання моделі: Flow ID, Source IP, Destination IP, Source Port, Destination Port, Timestamp

3. Виділення лише числових ознак

Для роботи алгоритмів машинного навчання були залишені тільки числові стовпці. Усі нечислові поля автоматично виключались.

4. Обробка пропущених і нескінченних значень

- Нескінченні значення (inf, -inf) замінено на NaN
- Усі NaN були замінені на середні значення відповідного стовпця

5. Масштабування ознак (нормалізація)

За допомогою StandardScaler з бібліотеки sklearn було проведено масштабування всіх числових ознак, щоб уникнути переважання однієї характеристики над іншими.

6. Кодування цільових міток (Label Encoding)

Оскільки колонка Label містила текстові значення (наприклад, "BENIGN", "DoS", тощо), її було перетворено на числові класи за допомогою LabelEncoder.

7. Балансування вибірки (SMOTE)

Через суттєву диспропорцію між нормальним і аномальним трафіком, було застосовано метод синтетичного балансування класів SMOTE, який дозволив покращити якість навчання моделі, зокрема для рідкісних типів атак.

Результатом цього етапу стало формування якісного, масштабованого і збалансованого набору даних, придатного для ефективного навчання моделі класифікації.

4.2 Побудова та навчання моделі

Після формування чистого, збалансованого та нормалізованого набору даних було здійснено побудову моделі машинного навчання код якої наведено в Додатку А, та для класифікації мережевого трафіку на нормальний (benign) і потенційно аномальний (DDos) трафік.

Серед популярних алгоритмів класифікації було обрано Random Forest Classifier — ансамблевий метод, заснований на побудові великої кількості дерев рішень з подальшим голосуванням більшості. Основні причини вибору саме цього алгоритму:

- висока стійкість до переобучення;
- здатність працювати з великою кількістю ознак;
- хороша інтерпретованість результатів (можна отримати важливість ознак);
- вбудована оцінка похибок та крос-валідація.

1. Для реалізації моделі використовувалась бібліотека Scikit-learn (sklearn.ensemble.RandomForestClassifier).

Перед тренуванням модель було навчено та протестовано за класичною схемою розбиття:

- тренувальна вибірка – 80% від усіх даних;
- тестова вибірка – 20% для оцінки якості моделі.

Для забезпечення рівномірного розподілу класів у вибірках використовувалась стратифікація (параметр stratify=y у функції train_test_split).

2. Для підвищення якості класифікації було використано GridSearchCV — метод повного перебору гіперпараметрів із крос-валідацією. Було протестовано комбінації таких параметрів:

- n_estimators: [100, 200, 300] — кількість дерев у лісі;
- max_depth: [10, 20, None] — максимальна глибина дерева;
- min_samples_split: [2, 5] — мінімальна кількість зразків для розбиття;
- min_samples_leaf: [1, 2] — мінімальна кількість зразків у листку;
- criterion: [‘gini’, ‘entropy’] — функція для оцінки якості поділу.

Модель RandomForestClassifier була навчена на тренувальному наборі даних після підбору найкращих параметрів. Час навчання склав приблизно X секунд (залежить від конфігурації обладнання та розміру вибірки).

Навчання здійснювалося з фіксованим random_state=42 для забезпечення відтворюваності результатів.

Після завершення тренування, фінальна модель була серіалізована та збережена у файл best_random_forest_model.pkl для подальшого використання. Це дозволяє не проводити повторне навчання при кожному запуску системи.

4.3 Перевірка працездатності моделі

Після завершення етапу навчання та тестування модель пройшла перевірку на працездатність у більш наближених до реальних умовах. Це дозволяє оцінити її придатність для подальшої інтеграції у систему моніторингу мережевого трафіку.

Для перевірки працездатності моделі був розроблений код який зображений у Додатку Б та була змодельована ситуація, коли логи надходять у вигляді окремих файлів або потоків даних. У рамках цієї перевірки виконувалися такі кроки:

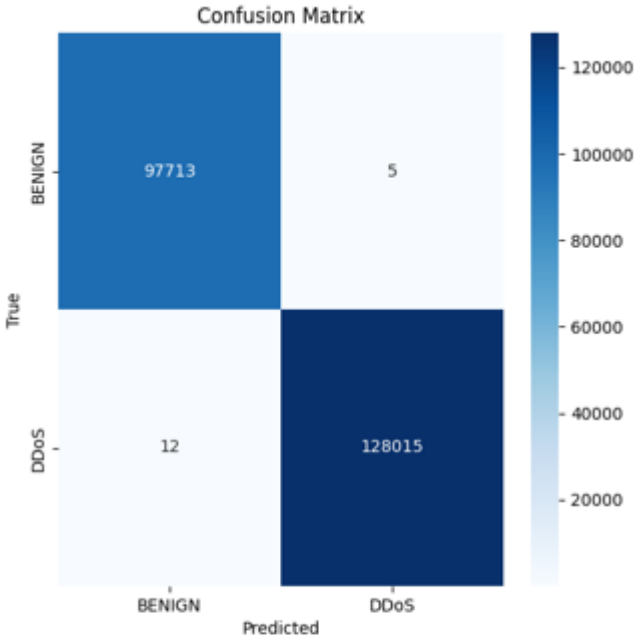
- Зчитування одного або кількох нових CSV-файлів, які не входили до навчальної чи тестової вибірки.
- Обробка даних за аналогічною схемою, яка використовувалась при тренуванні: видалення зайвих полів, обробка пропущених значень, нормалізація, кодування міток.
- Прогнозування класу для кожного запису з використанням вже навченої та збереженої моделі.
- Фіксація результатів класифікації, зокрема кількість виявлених аномалій та типи атак.

Важливою особливістю побудованої моделі є те, що вона навчалась виключно на прикладах DDoS-атак, тобто виконувалась бінарна класифікація: «нормальний трафік» (BENIGN) проти «DDoS-атака».

Таким чином, модель не має знань про інші типи аномалій (наприклад, PortScan, BruteForce, Infiltration), оскільки відповідні приклади були відфільтровані на етапі формування навчальної вибірки.

Це має суттєвий вплив на її поведінку в реальних умовах. Зокрема, якщо до моделі потрапить трафік з міткою PortScan, вона, найімовірніше, класифікує його як BENIGN, оскільки не розпізнає його як аномалію.

На (Рис.4.1) продемонстровано приклад правильного спрацювання моделі на записах із мітками DDoS. Модель коректно класифікувала ці записи як аномальні.



	precision	recall	f1-score	support
BENIGN	1.00	1.00	1.00	97718
DDoS	1.00	1.00	1.00	128027
accuracy			1.00	225745
macro avg	1.00	1.00	1.00	225745
weighted avg	1.00	1.00	1.00	225745

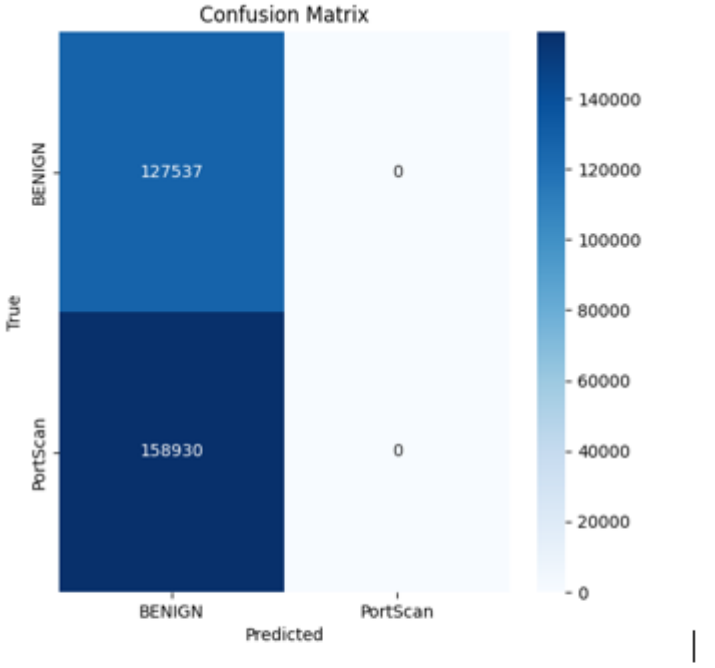
Рис. 4.1. Результат сканування логів на Ddos

Precision, Recall, F1-score для обох класів — 1.00 (або близько до 1.00) — це означає, що модель майже безпомилково класифікує як BENIGN, так і DDoS.

Accuracy = 1.00 — тобто майже всі приклади класифіковано правильно.

Загалом 17 помилок із 225745 прикладів — це дуже гарний результат (точність ≈ 99.99%).

На (Рис.4.2) продемонстровано приклад спрацювання моделі на записах із мітками PortScan. Модель класифікувала ці записи як звичайні логи.



	precision	recall	f1-score	support
BENIGN	0.45	1.00	0.62	127537
PortScan	0.00	0.00	0.00	158930
accuracy			0.45	286467
macro avg	0.22	0.50	0.31	286467
weighted avg	0.20	0.45	0.27	286467

Рис. 4.2. Результат сканування логів на PortScan

Precision для обох класів:

- BENIGN: 0.45 — це дуже низький результат, що означає, що тільки 45% класифікацій BENIGN є правильними.
- PortScan: 0.00 — модель не класифікує жоден випадок як PortScan правильно. Це величезна проблема.

Recall:

- BENIGN: 1.00 — модель правильно класифікує всі приклади класу BENIGN. Це дуже добре, але може бути ознакою перенавчання або того, що модель дуже домінує по класу BENIGN.
- PortScan: 0.00 — жоден приклад класу PortScan не був класифікований правильно.

F1-score:

- BENIGN: 0.62 — через низький precision, хоча recall 1.00, загальний F1-score для класу BENIGN не є дуже хорошим.
- PortScan: 0.00 — немає жодних правильних класифікацій для цього класу, тому F1-score = 0.

Accuracy: 0.45 — лише 45% усіх прикладів класифіковано правильно, що означає значний збій у роботі моделі.

Проаналізувавши матрицю, модель взагалі не виявляє PortScan, і всі приклади потрапляють в категорію BENIGN. Це є результатом того, що модель не навчена правильно відрізняти ці класи, та вона просто "відкидає" цей клас.

Висновки до розділу 4

У практичній частині реалізовано повний цикл побудови моделі машинного навчання для автоматичної класифікації мережевого трафіку. Основними етапами стали: очищення, нормалізація та балансування даних, побудова моделі Random Forest, її навчання з використанням GridSearchCV та подальше тестування. Результати експериментів показали, що модель ефективно і з високою точністю розпізнає DDoS-атаки, але демонструє низьку здатність виявляти інші типи атак, зокрема PortScan, через відсутність таких прикладів у навчальній вибірці. Це підкреслює важливість забезпечення репрезентативності даних під час навчання моделі. Збереження моделі у вигляді серіалізованого файлу дозволяє інтегрувати її у реальні системи моніторингу без повторного навчання.

Додатково було змодельовано реальний сценарій використання, коли логи надходять у вигляді нових, що дозволило перевірити стабільність і надійність моделі в динамічному середовищі. Отримані результати підтвердили практичну цінність розробленого підходу в задачах виявлення аномалій у мережевому трафіку.

ВИСНОВКИ

У ході проведеної роботи було здійснено глибоке занурення у світ інформаційної безпеки та аналізу подій у системах. Ми розглянули, наскільки важливим є розуміння загроз, правильна організація збору логів та вміння їх обробляти для своєчасного виявлення проблем.

Протягом дослідження було окреслено ключові принципи захисту інформації, виділено основні типи логів і методи роботи з ними, а також вивчено сучасні рішення, що дозволяють не лише реагувати на інциденти, а й запобігати їм заздалегідь. Значну увагу приділено аналізу методів агрегації та нормалізації логів, моніторингу в реальному часі, використанню систем типу SIEM для ефективного контролю та захисту інформаційної інфраструктури.

Окремо було проаналізовано роль логів у системах безпеки. Розглянуто основні типи журналів подій — логи периметрових пристроїв, подій Windows, кінцевих точок, додатків, проксі-серверів та пристроїв IoT. Визначено їхню структуру, особливості та значення для виявлення інцидентів безпеки.

Підсумовуючи, можна сказати, що управління логами — це не просто технічна задача, а справжнє мистецтво читати між рядків, розуміти поведінку системи та вчасно помічати перші ознаки небезпеки. У сучасному світі, де цінність даних зростає щодня, володіння такими знаннями і навичками є важливою складовою захисту будь-якої організації.

Загальний аналіз проведеного дослідження підтвердив, що комплексне використання логів, систем моніторингу та інструментів автоматизації є критично важливим для забезпечення стійкості інформаційних систем перед кіберзагрозами.

Отже, дослідження підтверджує актуальність і важливість комплексного підходу до захисту інформаційних систем шляхом належної роботи з логами, моніторингом та впровадженням передових технологій забезпечення інформаційної безпеки. Отримані результати можуть бути використані для

подальшого розвитку практик кібербезпеки та вдосконалення захисних механізмів в організаціях різного масштабу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is Information Security (InfoSec)? Microsoft Corporation. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-information-security-infosec#faqs

2. Guide to Computer Security Log Management, by Karen Kent and Murugiah Souppaya, National Institute of Standards and Technology (NIST) 2006. URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-92.pdf

3. About Event Logging. Microsoft Corporation. URL: https://learn.microsoft.com/en-us/windows/win32/eventlog/about-event-logging

4. Event Types. Microsoft Corporation. URL: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-types

5. Collecting and Analyzing Different Log Types. ManageEngine. URL: https://www.manageengine.com/log-management/siem/collecting-and-analysing-different-log-types.html

6. Winning the Cyber Arms Race With Machine Learning. SecurityWeek. URL: https://www.securityweek.com/winning-cyber-arms-race-machine-learning/

7. Enhancing Log Analysis with Machine Learning. Logit.io. URL: https://logit.io/blog/post/enhancing-log-analysis-with-machine-learning/#model-training

8. What is SIEM? IBM Corporation. URL: https://www.ibm.com/think/topics/siem

9. Еволюція стратегій кібербезпеки: роль SIEM та SOAR у захисті від сучасних загроз. Softico. URL: https://softico.ua/uk/knowledge/evolyutsiya-strategij-kiberbezpeki-rol-siem-ta-soar-u-zahisti-vid-suchasnih-zagroz/

10. What is Security Orchestration, Automation and Response (SOAR)? IBM Corporation. URL: https://www.ibm.com/think/topics/security-orchestration-automation-response

11. Data Aggregation in Cybersecurity. VPN Unlimited. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/data-aggregation?srsltid=AfmBOore0I_Mlb-bkKQ1ek67PvSFfy1PNsqgnIKfm5LlqFSg2pj1qF8p

12. Log Normalization. ManageEngine. URL: https://www.manageengine.com/products/eventlog/logging-guide/log-normalization.html

13. Log Correlation. ReasonLabs Cyberpedia. URL: https://cyberpedia.reasonlabs.com/EN/log%20correlation.html

14. What is Machine Learning? IBM Corporation. URL: https://www.ibm.com/think/topics/machine-learning

15. How Log Analysis is Evolving with AI and ML. Edge Delta. URL: https://edgedelta.com/company/blog/how-log-analysis-is-evolving-with-ai-and-ml

16. Log Monitoring Guide. Better Stack. URL: https://betterstack.com/community/guides/logging/log-monitoring/

17. What is Real-Time Monitoring? TechTarget. URL: https://www.techtarget.com/whatis/definition/real-time-monitoring

18. Log Monitoring: Benefits and Disadvantages. RevDeBug. URL: https://revdebug.com/blog/log-monitoring-benefits-disadvantages/

19. What is Log Analysis? Splunk. URL: https://www.splunk.com/en_us/blog/learn/log-analysis.html

20. Pattern Recognition. ReasonLabs Cyberpedia. URL: https://cyberpedia.reasonlabs.com/EN/pattern%20recognition.html

21. Guide to Computer Security Log Management. NIST. URL: https://csrc.nist.rip/library/NIST%20SP%20800-092%20Guide%20to%20Computer%20Security%20Log%20Management,%202006-09.pdf

22. What is the CIA Triad? Definition and Importance. Wallarm. URL: https://www.wallarm.com/what/cia-triad-definition

23. AUC-ROC Curve: Evaluating Performance of Classification Models in Machine Learning. Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

24. Confusion Matrix for Multi-Class Classification Problems. Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/

25. ROC and AUC – Evaluating a Classifier. Google Developers. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=uk

26. Pandas – Бібліотека для обробки та аналізу даних. Pandas Documentation. URL: https://pandas.pydata.org/

27. NumPy – Основна бібліотека для наукових обчислень. NumPy Documentation. URL: https://numpy.org/

28. Scikit-learn – Інструменти для машинного навчання в Python. Scikit-learn Documentation. URL: https://scikit-learn.org/stable/

29. Imbalanced-learn – Методи для роботи з незбалансованими даними (SMOTE тощо). Imbalanced-learn Documentation. URL: https://imbalanced-learn.org/stable/

30. Matplotlib – Бібліотека для створення графіків і візуалізацій. Matplotlib Documentation. URL: https://matplotlib.org/

31. Seaborn – Бібліотека для статистичної візуалізації даних. Seaborn Documentation. URL: https://seaborn.pydata.org/

32. Joblib – Інструмент для збереження моделей та об'єктів у Python. Joblib Documentation. URL: https://joblib.readthedocs.io/en/latest/

ДОДАТОК А – КОД ПРОГРАМИ ДЛЯ ГЕНЕРУВАННЯ МОДЕЛІ

```
import pandas as pd
import numpy as np
import os
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler, LabelEncoder
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
import seaborn as sns
import joblib

# 1. Папка з логами
folder_path = 'Посилання на папку!!!'
csv_files = [os.path.join(folder_path, f) for f in os.listdir(folder_path) if f.endswith('.csv')]

if not csv_files:
    raise FileNotFoundError("❌ CSV-файли не знайдено в папці.")

print(f"Знайдено {len(csv_files)} файлів.")
dataframes = []

for file in csv_files:
    try:
        print(f"Завантаження: {file}")
```

```
        temp_df = pd.read_csv(file, encoding='utf-8', low_memory=False)
    except UnicodeDecodeError:
        temp_df = pd.read_csv(file, encoding='ISO-8859-1', low_memory=False)

    temp_df.columns = temp_df.columns.str.strip()  # прибираємо пробіли
    if 'Label' in temp_df.columns:
        dataframes.append(temp_df)
    else:
        print(f"⚠ Пропущено (немає 'Label'): {file}")

# 2. Об'єднання даних
df = pd.concat(dataframes, ignore_index=True)

# 3. Перевірка наявності мітки
label_col = 'Label'
if label_col not in df.columns:
    raise KeyError(f"Стовпець '{label_col}' не знайдено.")

# 4. Виділення ознак і міток
X = df.drop(columns=['Flow ID', 'Source IP', 'Source Port', 'Destination IP', 'Destination
Port', label_col], errors='ignore')
y = df[label_col]

# 5. Тільки числові стовпці
X = X.select_dtypes(include=[np.number])
X.replace([np.inf, -np.inf], np.nan, inplace=True)
X.fillna(X.mean(), inplace=True)

# 6. Масштабування
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)

# 7. Кодування міток
le = LabelEncoder()
y_encoded = le.fit_transform(y)

# 8. Розділення на тренувальні та тестові
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.3, random_state=42)

# 9. Балансування класів SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# 10. Параметри для GridSearch
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2]
}

# 11. Навчання моделі
rf = RandomForestClassifier(random_state=42)
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, n_jobs=-1, verbose=1)
grid_search.fit(X_train_resampled, y_train_resampled)

# 12. Найкраща модель
best_model = grid_search.best_estimator_
```

```
# 13. Збереження моделі
joblib.dump(best_model, 'best_random_forest_model.pkl')
print("✅ Модель збережена як 'best_random_forest_model.pkl'")

# 14. Cross-validation
cross_val_scores = cross_val_score(best_model, X_train_resampled, y_train_resampled, cv=5)
print("Cross-validation scores:", cross_val_scores)

# 15. Оцінка на тестовому наборі
y_pred = best_model.predict(X_test)
print("\n=== Classification Report ===")
print(classification_report(y_test, y_pred, target_names=le.classes_))
print("\n=== Confusion Matrix ===")
cm = confusion_matrix(y_test, y_pred)
print(cm)

# 16. Візуалізація
plt.figure(figsize=(6, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=le.classes_, yticklabels=le.classes_)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix (Random Forest)')
plt.show()
```

ДОДАТОК Б – КОД ПРОГРАММИ ДЛЯ ТЕСТУВАННЯ МОДЕЛІ

```
import pandas as pd
import numpy as np
import joblib
import os
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

model_path = "'Посилання на модель!!!"
if not os.path.exists(model_path):
    raise FileNotFoundError(f"Модель не знайдено: {model_path}")
model = joblib.load(model_path)
print("✅ Модель завантажена")

file_path = "E'Посилання на папку!!!"
df = pd.read_csv(file_path, encoding="ISO-8859-1")  # Змінити кодування, якщо треба
df.columns = df.columns.str.strip()
print("✅ Логи завантажені")

if 'Label' not in df.columns:
    raise KeyError("Колонку 'Label' не знайдено для порівняння!")

X = df.drop(columns=['Flow ID', 'Source IP', 'Source Port', 'Destination IP',
                     'Destination Port', 'Timestamp', 'Label'], errors='ignore')
y = df['Label']
```

```
X = X.select_dtypes(include=[np.number])
X = X.replace([np.inf, -np.inf], np.nan)
X.fillna(X.mean(), inplace=True)

# Масштабування
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

le = LabelEncoder()
y_encoded = le.fit_transform(y)

y_pred = model.predict(X_scaled)

print("\n=== Classification Report ===")
print(classification_report(y_encoded, y_pred, target_names=le.classes_))

print("\n=== Confusion Matrix ===")
cm = confusion_matrix(y_encoded, y_pred)
print(cm)

# Візуалізація
plt.figure(figsize=(6, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=le.classes_,
yticklabels=le.classes_)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Поняття інформаційної безпеки: Складові

Інформаційна безпека (InfoSec) – це набір процедур та інструментів, які захищають усю делікатну корпоративну інформацію від неправомірного використання, несанкціонованого доступу, псування або знищення. InfoSec включає інформаційну тріаду де фігурують Конфіденційність, Цілісність, Доступність



Рис. 1. Інформаційний трикутник

Конфіденційність

Доступ до інформації лише авторизованим користувачам

Цілісність

Інформація не змінена та не пошкоджена несанкціоновано

Доступність

Вчасний та надійний доступ до даних для уповноважених осіб

4

Типи логів та їх структура

Perimeter Logs - фіксують події на брандмауерах, VPN, IDS/IPS; важливі для виявлення та спроб вторгнення.

```
2015-07-06 11:35:26 ALLOW TCP 10.40.4.182
10.40.1.11 63064 135 0 - 0 0 0 - - - SEND
```

Windows Event Logs - події ОС, безпека, автентифікація, помилки.

```
Warning 4/28/2020 12:32:47 PM
WLAN-AutoConfig 4003 None
```

Application Logs - збої додатків, запити до баз даних, підозріла активність.

```
02-AUG-2013 17:38:48 * (CONNECT_DATA=
(SERVICE_NAME=dev12c)(CID=(PROGRAM=sqlplu
s)(HOST=oralinux1)(USER=oracle))) *
(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.2.12
```

Endpoint Logs - знімні носії, драйвери, дії користувачів.

```
Error 6/20/2019 5:00:45 PM Terminal
Services- Printers 1111 None
```

IoT Logs - журнали з пристроїв Інтернету речей, що фіксують технічний стан і мережеві з'єднання.

5

Методи обробки логів

Агрегація

Агрегація даних — це процес збору, узгодження та інтеграції інформації з різних джерел для створення єдиного, узагальненого представлення даних

Нормалізація

Нормалізація журналів (логів) — це процес перетворення даних у єдиний, стандартизований формат, що забезпечує їхню узгодженість та зручність для аналізу.

Відобрадення логу до обробки	127.0.0.1 user-identifier andy [12/Nov/2021:20:25:11-0700] "GET /apache_pb.gif HTTP/1.0"
Відобрадення логу після оброки	"127.0.0.1" refers to the client's IP address when a server request is made. "user-identifier" refers to the client's identification protocol. "andy" refers to the user ID. "[12/Nov/2021:20:25:11-0700]" refers to the date and time the

6

Кореляція логів - це

Процес збору та аналізу даних журналу з різних джерел для виявлення потенційних порушень безпеки. Це включає агрегування та нормалізацію логів з різних систем, таких як антивірусне програмне забезпечення, брандмауери та мережеві пристрої

Кореляція логів складається з двох основних етапів: збору та аналізу. Першим кроком є збір усіх даних журналу та створення центрального сховища

Другим кроком є аналіз цих логів, щоб виявити будь-які потенційні зв'язки чи кореляції. Програмне забезпечення загалом спрощує процес кореляції завдяки системі, призначеній для позначення певних послідовностей або порушень

У той же час кореляція логів також може надати історичний контекст для порушень, надаючи командам необхідну перспективу того, як порушення могло статися.

7

Розпізнавання образів

Розпізнавання образів — це підгалузь штучного інтелекту, яка передбачає ідентифікацію шаблонів і тенденцій на основі даних. Він дозволяє автоматично класифікувати дані на основі попередньо встановлених правил і визначень.

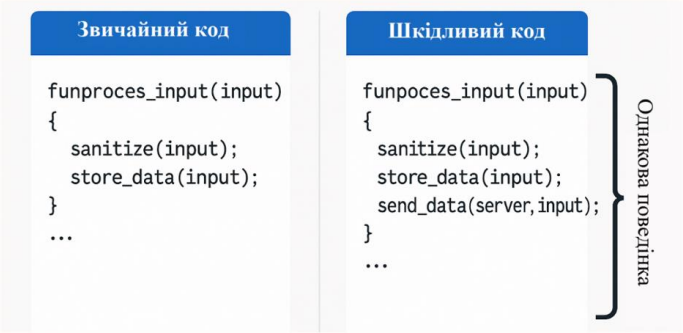


Рис. 2. Порівняння двох частин коду

8

Виявлення ефективності AI або ML моделі

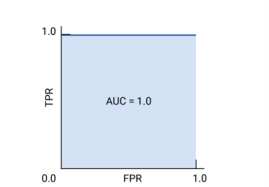


Рис. 3. Неможлива або ідеальна крива

ROC-крива з AUC = 1 (ідеальний класифікатор)

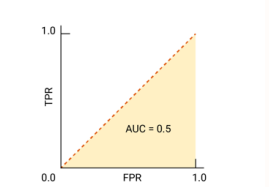


Рис. 4. Випадкова крива

ROC-крива з AUC = 0.5 (випадковий класифікатор)

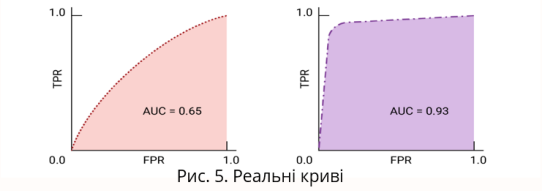


Рис. 5. Реальні криві

ROC-крива з AUC між 0.5 і 1 (ефективний, але не ідеальний класифікатор).

9

Чутливість і специфічність

Матриця плутанини – це інструмент візуалізації, який допомагає зрозуміти ефективність моделі класифікації. Це таблиця, яка підсумовує прогнози, зроблені моделлю, порівняно з фактичними значеннями істини

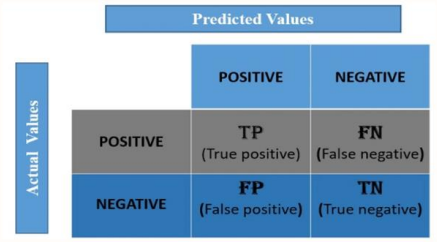


Рис. 6. Матриця плутанини

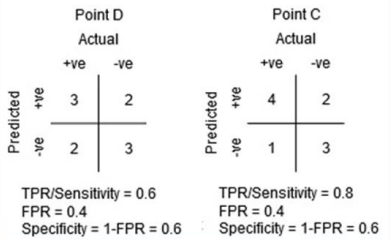


Рис. 7. Приклад використання матриці

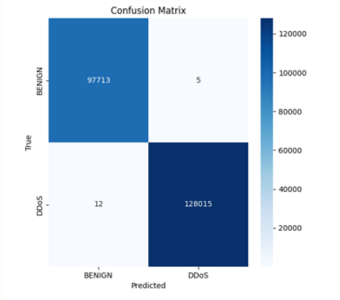
10

Етапи розробки та впровадження проєкту:

1. Визначення об’єкту, предмету, мети та завдань дослідження.
2. Збір та аналіз літератури.
3. Аналіз впливу атак на відмову в обслуговуванні на корпоративні мережі
4. Аналіз сучасних методів моніторингу логів інформаційних систем
5. Дослідження підходів штучного інтелекту та машинного навчання для обробки моніторингових логів
6. Розробка застосунку для аналізу логів на основі сучасного методу машинного навчання

11

Результат сканування логів на Ddos



	precision	recall	f1-score	support
BENIGN	1.00	1.00	1.00	97718
DDoS	1.00	1.00	1.00	128027
accuracy			1.00	225745
macro avg	1.00	1.00	1.00	225745
weighted avg	1.00	1.00	1.00	225745

Рис. 6.

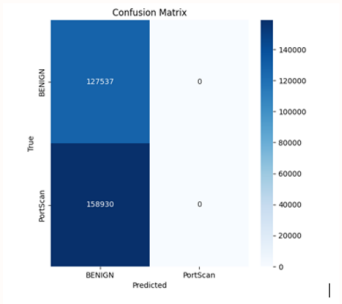
Precision, Recall, F1-score для обох класів — 1.00 (або близько до 1.00) — це означає, що модель майже безпомилково класифікує як BENIGN, так і DDoS.

Accuracy = 1.00 — тобто майже всі приклади класифіковано правильно.

Загалом 17 помилок із 225745 прикладів — це дуже гарний результат (точність ≈ 99.99%).

12

Результат сканування логів на PortScan



	precision	recall	f1-score	support
BENIGN	0.45	1.00	0.62	127537
PortScan	0.00	0.00	0.00	158930
accuracy			0.45	286467
macro avg	0.22	0.50	0.31	286467
weighted avg	0.20	0.45	0.27	286467

Рис. 7.

PortScan: для Precision, Recall, F1-scor = 0

Це дуже низький результат, модель не класифікує жоден випадок як PortScan правильно. Це величезна проблема.

BENIGN: для Precision, Recall, F1-scor = 0.45, 1, 0.62

Це низький результат, що означає, що тільки 45% класифікацій BENIGN є правильними.