Міністерство освіти і науки України
Державний університет «Житомирська політехніка»

А. А. Єфіменко, І. Г. Фальковський

ОСНОВИ СИСТЕМНОГО ТА МЕРЕЖНОГО МОНІТОРИНГУ

Навчальний посібник

Житомир
2025

УДК 004.45:004.42

Г61

Рекомендовано до друку Вченою радою
Державного університету “Житомирська політехніка”,
протокол № 12 від 01 вересня 2025 року.

Рецензенти:

Бабенко В. Г. – доктор технічних наук, професор, професор кафедри інформаційної безпеки та комп’ютерної інженерії Черкаського державного технологічного університету;

Дорогий Я. Ю. – доктор технічних наук, професор, завідувач кафедри прикладної математики та інформатики Донецького національного технічного університету;

Воротніков В. В. – доктор технічних наук, доцент, професор кафедри комп’ютерної інженерії та кібербезпеки Державного університету “Житомирська політехніка”.

Єфіменко А.А., Фальковський І. Г.

Г61 Основи системного та мережного моніторингу: навч. посібн. / А.А.Єфіменко, І. Г. Фальковський. Електронні дані. – Житомир: «Житомирська політехніка», 2025. – 224 с.

ISBN 978-966-683-617-8

Навчальний посібник охоплює принципи організації спостереження за IT-системами та мережами з метою забезпечення їхньої стабільної роботи, безпеки та продуктивності. Вивчаються архітектура моніторингових систем, методи збору та аналізу системних і мережн'их метрик, протоколи SNMP, NetFlow, sFlow, інструменти Nagios, Zabbix, Grafana, Wazuh тощо. Розглядаються питання візуалізації даних, виявлення аномалій, інтеграції з системами безпеки та реагування на загрози.

Навчальний посібник призначено для здобувачів вищої освіти, які навчаються за спеціальностями F7 Комп’ютерна інженерія та F5 Кібербезпека.

В оформленні обкладинки використано комікс із фразами *«¡El perro es redondo! ¡El perro es plano!»* поширюється в інтернеті щонайменше з 2016 року, здебільшого в іспаномовному середовищі. Автор зображення наразі невідомий — найраніше зафіксоване використання датовано 2018 роком на форумі Antronio.cl. Якщо ви володієте інформацією про його походження, будемо вдячні за уточнення.

УДК 004.45:004.42

Навчальне видання
Єфіменко Андрій Анатолійович, Фальковський Ігор Геннадійович
Навчальний посібник
Електронне видання
Комп’ютерний дизайн та верстка: Фальковський І.Г.
Державний університет «Житомирська політехніка»
вул. Чуднівська, 103, м. Житомир, 10005

© Фальковський І.Г., 2025

ISBN 978-966-683-617-8

Зміст

Передмова 5
Розділ 1. Вступ до системного моніторингу 5
1. Поняття моніторингу та системного моніторингу 5
2. Визначення системного моніторингу та його роль у сучасних інформаційних технологіях 10
3. Основні принципи та цілі системного моніторингу 12
4. Види метрик, що вимірюються при системному моніторингу 15
Висновки до розділу 23
Контрольні запитання 23
Розділ 2. Архітектура системного моніторингу 25
1. Компоненти системи моніторингу: агенти, сервери, бази даних 25
2. Розгорнуті та розподілені архітектури моніторингу. 33
3. Протоколи зв'язку між компонентами системи моніторингу 36
Висновки до розділу 41
Контрольні запитання 42
Розділ 3. Вступ до мережного моніторингу 43
1. Визначення мережного моніторингу 43
2. Пояснення основних понять та термінів 44
3. Роль мережного моніторингу у сучасних IT системах 46
4. Оптимізація роботи мережі 46
5. Забезпечення стабільності та швидкодії 49
6. Стратегії оптимізації для забезпечення стабільності та швидкодії мережі 50
7. Роль мережного моніторингу в виявленні проблем. 54
8. Виявлення типових проблем через мережний моніторинг 55
9. Забезпечення безпеки мережі 57
10. Системи моніторингу трафіку 59
11. Переваги використання сучасних технологій 63
Висновки до розділу 66
Контрольні запитання 67
Розділ 4. Протоколи збору трафіку 68
1. Вступ до протоколів збору трафіку 68
2. Протокол збору трафіку SNMP 69
3. Протокол збору трафіку IPFIX 75
4. Протокол збору трафіку JFlow 77
5. Визначення та основні характеристики протоколу NetFlow 80
6. Походження та розвиток sFlow 84
7. Використання протоколів NetFlow та sFlow для збору мережних статистичних даних 88
Висновки до розділу 89
Контрольні запитання 90

Розділ 5. Інструменти та технології системного моніторингу 91
1. Огляд популярних інструментів системного моніторингу. Nagios. 91
2. Огляд інструменту системного моніторингу Zabbix. 100
3. Nagios vs Zabbix 108
4. Огляд Prometeus 109
5. Grafana 118
6. Практичне використання інструментів для моніторингу 121
7. Налаштування сповіщень та автоматизація управління системним моніторингом 123
Висновки до розділу 125
Контрольні запитання 126
Розділ 6. Аналіз та візуалізація даних мережного моніторингу. 127
1. Методи аналізу даних мережного моніторингу та їх застосування для виявлення аномалій та проблем в мережі. 127
2. Типи інструментів моніторингу мережі 147
3. Використання інструментів візуалізації даних для створення графіків, діаграм та звітів. 153
4. Розробка звітів та дашбордів для ефективного моніторингу мережі 159
5. Глобальні дашборди кіберзагроз. 164
Висновки до розділу 165
Контрольні запитання 166
Розділ 7. Впровадження систем моніторингу та безпеки IT-інфраструктури 166
1. Огляд Wazuh як відкритої платформи для моніторингу безпеки 166
2. Огляд Snort як системи виявлення та запобігання вторгненням 179
3. Огляд Splunk Enterprise Security 183
4. Огляд IBM QRadar: архітектура та можливості 185
5. Огляд Elastic Security. 191
6. Огляд MS SCOM як інструменту системного моніторингу 195
Висновки до розділу 202
Контрольні запитання 203
Розділ 8. Системи моніторингу та стратегії реагування на інциденти кібербезпеки .. 205
1. Застосування моніторингу для виявлення загроз безпеці. 205
2. Використання системного та мережного моніторингу для виявлення вторгнень, зламів та інших загроз безпеці. 215
3. Аналіз логів, виявлення аномалій та паттернів, пов'язаних із зловмисними діями. 222
4. Розробка стратегій реагування та відновлення після виявлення загроз. 228
5. Компроміси безпеки 237
Висновки до розділу 244
Контрольні запитання 245
Джерела та посилання 246

Передмова

У навчальному посібнику «Основи системного та мережного моніторингу» розглянуто базові теоретичні та прикладні аспекти побудови, функціонування та впровадження систем моніторингу в інформаційних інфраструктурах. Основна мета цього видання — сформувати в здобувачів освіти цілісне уявлення про принципи системного та мережного моніторингу, його значення для забезпечення стабільної роботи та безпеки IT-середовища.

Матеріал посібника структуровано за тематичними розділами, які охоплюють широке коло питань — від основних понять і архітектур моніторингових систем до використання сучасних інструментів і технологій для збору, аналізу та візуалізації даних. Особливу увагу приділено протоколам збору трафіку (SNMP, NetFlow, sFlow, IPFIX та ін.), практичному застосуванню систем на кшталт Zabbix, Prometheus, Grafana, Splunk та Wazuh, а також використанню моніторингу як інструменту виявлення загроз і забезпечення кібербезпеки.

Описані у посібнику технології та засоби дозволяють не лише глибше зрозуміти принципи функціонування сучасних інформаційних систем, а й засвоїти підходи до створення ефективної інфраструктури моніторингу, здатної виявляти та реагувати на потенційні загрози в режимі реального часу.

Кожний розділ посібника супроводжується контрольними питаннями для перевірки та самоперевірки в здобувачів освіти рівня засвоєння матеріалу. Наприкінці розділів пропонується перелік рекомендованих джерел для всіх, хто потребує глибшого й детальнішого опрацювання відповідних тем.

Оскільки у даному виданні висвітлено передусім теоретичні питання, наголошуємо на важливості доповнення викладеного тут матеріалу серією лабораторних і/або практичних робіт, у межах яких здобувачі освіти мали б змогу поглибити своє розуміння теоретичних основ та здобути необхідні практичні навички з основ системного та мережного моніторингу, розгортання та налаштування систем моніторингу.

Посібник орієнтовано на здобувачів вищої освіти спеціальностей F5 Кібербезпека та захист інформації та F7 Комп’ютерна інженерія, однак його зміст буде корисним і для студентів інших IT-спеціальностей, зокрема, F2 Інженерія програмного забезпечення, F3 Комп’ютерні науки, F6 Інформаційні системи та технології, а також спеціальності G5 Електроніка, електронні комунікації, приладобудування та радіотехняка. Видання також може бути корисним для практиків, які прагнуть систематизувати знання в галузі моніторингу IT-інфраструктур.

Сподіваємося, що цей посібник стане надійною основою для подальшого професійного розвитку майбутніх фахівців у сфері інформаційних технологій та кібербезпеки.

Розділ 1. Вступ до системного моніторингу

1. Поняття моніторингу та системного моніторингу

Будь-яка людина, пов’язана з інформаційними технологіями знає, що моніторинг є важливим. Іноді — дуже важливим. Є, звичайно, крайність, коли

фахівець сподівається дізнатися про аварійну ситуацію за дзвінками розлючених клієнтів, але ця крайність швидко переходить або на розуміння попередньої фрази, або на швидке розлучення з цим фахівцем. Можу провести аналогію: «Або системний адміністратор використовує моніторинг роботи серверів та мереж, або у нього з'являється дуже багато вільного часу та дуже мало грошей». Тож зупиняємось на тому, що моніторинг має бути. Але тут є безліч розгалужень у виборі підходу до створення системи моніторингу, її архітектури, вартості ну і нарешті глибини. І ось ми наблизилися до поняття що ж таке той самий моніторинг, які його види існують, на яких принципах він будується.

Рис. 1.1. «Спостерігачі» на «об’єкті моніторингу»

Зверніть увагу на ситуацію на картинці з двома «спостерігачами» та «об’єктом моніторингу» (рис.1.1.). На жаль першоджерело цього коміксу не вдалося знайти – власники телеграм каналів люблять щось поцупити без згадки автора. Комікс із фразами «¡El perro es redondo! ¡El perro es plano!» поширюється в інтернеті щонайменше з 2016 року, здебільшого в іспаномовному середовищі. На рис. 1.1 спостерігачі живуть на об’єкті, але через неправильно налаштований моніторинг, чи скоріше його відсутність, мають хибне та часткове уявлення про своє місце проживання☺ чи у випадку системного адміністратора – місце своєї роботи.

Якщо почали з коміксу, буде доречним навести ще ілюстрацію концепції світу, що стоїть на слонах і черепасі. Вона належить до міфологічних уявлень різних культур. Найбільш відомий її варіант походить з давньоіндійської міфології, де світ зображується як диск, який тримають слони, що стоять на спині велетенської черепахи (а іноді й на спині змії або дракона). Зображення на рис. 1.2 є малюнком-ілюстрацією 1877 року до європейських філософських текстів, які популяризували ці міфи. Зображення такого типу часто публікували в енциклопедіях чи книгах XVIII–XIX століть.

Ця міфологічна концепція, хоча й далека від сучасного наукового розуміння, добре ілюструє важливість опорних структур, які забезпечують стабільність системи. Подібні принципи ми можемо побачити і в організації системного моніторингу.

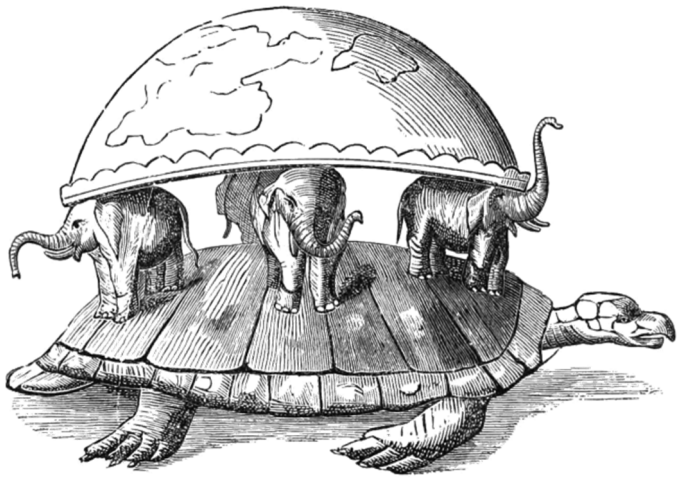


Рис. 1.2. Малюнок 1877 року, що зображує світ, який тримається на спинах чотирьох слонів, що стоять на спині черепахи.

Подібно до того, як на зображенні світ тримається на черепасі та слонах, система моніторингу має бути побудована на надійних «опорах» — правильно обраних точках спостереження та ключових елементах (items). Проте важливо пам'ятати, що точка спостереження визначає наше сприйняття. Спостерігачі, які знаходилися 'всередині' системи, не могли побачити її цілком, що й призвело до помилкового уявлення про будову світу. Аналогічно, вибір неправильних або обмежених точок спостереження у системі моніторингу може створити хибне уявлення про її стан. Тому потрібно забезпечувати різнобічний та актуальний погляд на всю систему, адаптуючи моніторинг до змін у середовища.

Якщо відкрити статтю моніторинг у Вікіпедії, то прочитаємо, що моніторинг або моніторинґ (укр. спостереження) — система постійного спостереження за явищами і процесами, що проходять в навколишньому середовищі, суспільстві, результати якого слугують для обґрунтування управлінських рішень по забезпеченню безпеки людей та об'єктів/суб'єктів (наприклад, економіки).

У широкому сенсі моніторинг — це процес безперервного або регулярного (періодичного) збору інформації про стан певних параметрів об'єкту або суб'єкту спостереження (моніторингу).

Метою моніторингу може бути накопичення інформації для її подальшого аналізу та прийняття управлінського рішення, або постійне відслідковування стану об'єкту моніторингу без збереження попередньої інформації про об'єкт з метою своєчасного реагування (прийняття управлінського рішення) при певних кількісних або якісних змінах об'єкта.

Моніторинг є ключовою складовою управління та підтримки інформаційних систем та технологій. Існує кілька видів моніторингу, кожен з яких має свої характеристики та важливість для специфічних цілей та потреб [4], [42].

Здійснення моніторингу може бути автоматизоване: при автоматичному контролі відбувається отримання і обробка інформації про стан об'єкта/суб'єкта та зовнішніх умов для виявлення подій, що визначають управлінські дії. Такою подією може бути будь-який заздалегідь заданий параметр: поява деталі з

розмірами, що виходять за допустимі межі, коротке замикання електричної мережі, вихід температури за встановлене значення, аварія обладнання та інші.

Цей навчальний предмет не охоплює питання моніторингу метеорологічних явищ чи медичних показників, таких як артеріальний тиск або рівень насичення крові киснем у хворих на COVID-19, хоча останніми роками ці напрями набули значної актуальності. Дисципліна орієнтована саме на вивчення системного та мережного моніторингу. Водночас у межах сучасних інформаційних систем існує значна кількість різновидів моніторингу, які об’єднуються під загальною назвою — інфраструктурний моніторинг (Infrastructure Monitoring). Основні види моніторингу інформаційних систем та їх характеристики наведено в табл. 1.1.

Таблиця 1.1

Види моніторингу

Вид моніторингу	Опис	Ціль моніторингу
Системний моніторинг (System Monitoring)	Відстеження параметрів та показників, що стосуються апаратного та програмного забезпечення системи (операційна система, процесор, пам'ять, мережа тощо)	Оцінка продуктивності, надійності та ефективності системи, виявлення аномалій та проблем для швидкого реагування
Мережний моніторинг (Network Monitoring)	Моніторинг мережних компонентів, трафіку та з'єднань, щоб виявити незвичайну або неправильну активність	Забезпечення безпеки, виявлення і вирішення проблем у мережній інфраструктурі
Дода́тковий моніторинг (Application Monitoring)	Спостереження за додатками, оцінка їх продуктивності, надійності та відгуку на дії користувачів	Оцінка роботи дода́тків, виявлення та усунення проблем, покращення відповідності до вимог користувачів
Безпековий моніторинг (Security Monitoring)	Спостереження за подіями та активністю, що можуть вказувати на потенційні кіберзагрози та порушення безпеки	Забезпечення безпеки, виявлення та запобігання кібератак, захист від несанкціонованого доступу.
Журналювання (Logging)	Запис подій, дій та стану системи та додатків для подальшого аналізу та виявлення аномалій	Забезпечення можливості аналізу та виявлення незвичайної активності, вирішення проблем та безпеки

Різниця між цими видами моніторингу полягає в тому, що кожен з них спрямований на відстеження та аналіз певних аспектів системи та її екосистеми. Системний моніторинг орієнтований на апаратну та програмну частину системи, мережний - на мережну інфраструктуру, додаткОвий - на програмні додАтки, безпековий - на виявлення загроз та захист від них, а журналювання - на реєстрацію та збереження подій для аналізу. Ці види моніторингу є однаково важливими для забезпечення ефективності, надійності та безпеки систем та додатків.

Системний моніторинг - це комплексний процес автоматичного або ручного нагляду, аналізу та оцінки параметрів та функцій певної системи або групи систем. Цей процес організується для надання повної та об'єктивної інформації про стан системи в реальному часі або на певних етапах її роботи. Основна мета

- забезпечити вчасне виявлення та реагування на проблеми або відхилення в роботі системи для забезпечення її стабільності та ефективності.

Для кращого розуміння, що таке сучасний системний моніторинг необхідно коротко розглянути його еволюцію в сучасному світі IT. А на рис 1.3 – показана аналогія, що виникає при згадці поняття «Еволюція».

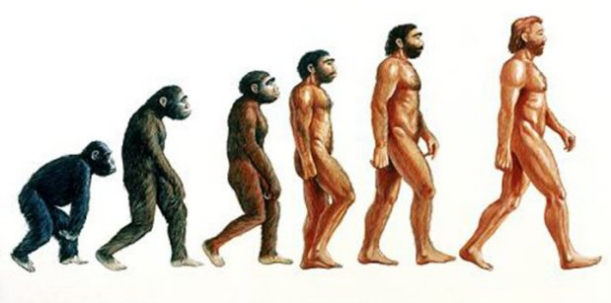


Рис. 1.3. Поняття «Еволюція»

- ***Період зародження.*** Початково системний моніторинг був обмеженим та базувався на мінімальних засобах. Основний фокус був на моніторингу апаратного забезпечення (процесор, пам'ять, диск) та базових операційних параметрів, що забезпечувалося системою агентів. Агент - це програмний компонент, що встановлюється на контрольованому об'єкті (системі, сервері, мережному пристрої), щоб збирати та передавати метрики про його стан. Такий підхід вимагав великого обсягу адміністративної роботи для управління та підтримки системи. Пізніше буде розглянуто, і не тільки теоретично, поняття агентів моніторингу, метрик та інші, що зустрічаються у цій лекції.
- ***Ера мереж та комп'ютерних систем.*** З поширенням мереж та зростанням складності систем з'явилися інструменти для моніторингу мережної активності та обміну даними між системами. Розвиток агентної та безагентної (agentless) архітектур для збору метрик на першому етапі призвів до появи централізованих систем управління агентами, що дозволяли керувати та оновлювати їх масово. Звичайно, що використання агентів дає величезні бонуси завдяки глибокій інтеграції та доступу до детальної інформації на рівні кожної системи, а також можливість використання локальних ресурсів для обробки та аналізу даних. Є і мінуси, а саме велика кількість агентів може призвести до додаткового навантаження на систему та зменшення продуктивності.

У випадку використання агент-лесс архітектури, збір та аналіз метрик відбувається без встановлення спеціальних агентів на кожному об'єкті. Замість цього, дані можуть збиратися з централізованих джерел або використовуватися агенти, які вже присутні на об'єкті (наприклад SNMP (Simple Network Management Protocol) та WMI (Windows Management Instrumentation) протоколи, що використовуються для моніторингу та керування комп'ютерними системами та мережами) [60], [22].

- ***Сучасні тенденції.*** Зростання величини даних і обчислювальної потужності призвело до необхідності розвивати системи моніторингу для масштабованості. Поява хмарних технологій призвела до необхідності

моніторингу ресурсів у віртуальних оточеннях. Автоматизація та використання штучного інтелекту для прогнозування та управління ресурсами [22], [30].

- ***Зосередження на бізнес-метриках.*** Замість виключно технічних параметрів ставиться акцент на метриках, що впливають на бізнес-показники. А саме враховуються показники доступності для кінцевих користувачів і взаємозв'язку з фінансовими результатами.
- ***Розширення Функціональності.*** Розвиток інтеграції з іншими системами, такими як системи логування, моніторинг безпеки, та іншими елементами IT-екосистеми. Аналіз даних для прогнозування та попередження проблем заздалегідь.

Зараз системний моніторинг - це не просто інструмент для виявлення технічних неполадок, але і стратегічний елемент, що сприяє покращенню ефективності, забезпеченню надійності, та відповідності бізнес-цілям компанії.

2. Визначення системного моніторингу та його роль у сучасних інформаційних технологіях

Зупинимось на ролі системного моніторингу у сучасних інформаційних технологіях, а саме на значенні системного моніторингу для бізнесу та організацій.

Системний моніторинг є ключовим елементом в управлінні та забезпеченні ефективності інформаційних технологій (IT) в сучасних бізнес-середовищах. Цей процес включає систематичний нагляд за функціональністю, продуктивністю та надійністю IT-систем, дозволяючи забезпечити безперебійну роботу інфраструктури та уникнути багатьох проблем для бізнесу за порівняно невеликі кошти [22], [30].

Уявімо, що Ви оволоділи навичками системного моніторингу, відчуваєте наснагу до узагальнення отриманих метрик та бажання оптимізувати за допомогою системного моніторинг інфраструктуру свого підприємства, але на Ваш жах на підприємстві він відсутній як поняття. Ну може, як залишок потуг попереднього персоналу десь «крутиться» покритий мохом сервер MRTG або покритий кактусами сервер Cacti. Це згадка про дві популярні на початку 2000-х системи вимірювання завантаженості та доступності каналів зв’язку. Їх широко використовували для перевірки т.з. білінгу з провайдером каналу зв’язку.

Це – лірика, хоча і дуже сумна і може з Вами трапитися. Які ж будуть Ваші аргументи бізнес керівництву для розгортання чи удосконалення системного моніторингу:

- Забезпечення безперебійності роботи.
  - Моніторинг дозволяє оперативно виявляти та вирішувати технічні проблеми, запобігаючи перебоям у роботі систем та сервісів.
  - Передбачення можливих збоїв та їх усунення до того, як вони вплинуть на бізнес-процеси.
- Підвищення продуктивності.
  - Визначення слабких місць у системі та оптимізація ресурсів для максимальної продуктивності.

- ❖ Аналіз метрик дозволяє ефективно розподіляти ресурси та уникати перевантажень.
- ➢ Зменшення витрат та підвищення вартості.
  - ❖ Ефективний моніторинг може допомогти уникнути непланованих витрат на ремонт або відновлення після аварій.
  - ❖ Оптимізація роботи системи дозволяє зменшити витрати на енергію та ресурси.

Необхідно чітко розуміти, що стратегія IT та системний моніторинг тісно пов'язані між собою і взаємодіють для підвищення ефективності роботи організації та досягнення її бізнес-цілей [30], [4]. Саме ця взаємодія є вагомим аргументом для керівництва щодо доцільності впровадження систем моніторингу. У цьому контексті важливо підкреслити, що впровадження моніторингу не обмежується лише контролем стану систем, а забезпечує організації низку додаткових переваг, які безпосередньо впливають на її розвиток та конкурентоспроможність.

- ➢ Вирішення стратегічних завдань.
  Системний моніторинг адаптується під стратегічні цілі IT, спрямовуючи свою діяльність на вимірювання параметрів, які важливі для стратегічного успіху. Як найпростіший приклад наводимо прогнозування витрат ресурсів. Вимірювання витрат ресурсів (процесор, пам'ять, мережа) дозволяє адекватно вирахувати, які ресурси будуть потрібні для забезпечення планової доступності при зростанні користувацького попиту.
- ➢ Визначення ключових показників ефективності.
  Метрики, які збираються системним моніторингом, стають основою для визначення ключових показників ефективності (KPI - Key Performance Indicator) IT-стратегії. Приклад: підвищення ефективності обробки транзакцій у фінансовому додатку -визначаємо навантаження на складові фінансового додатку (application, базу даних, WEB-сервер) і у найпростішому випадку перерозподіляємо ресурси.
- ➢ Прискорення прийняття рішень.

Моніторинг надає звіти у реальному, або близькому до реального часі, які допомагають приймати обґрунтовані стратегічні рішення. Тут навіть не потрібно наводити приклади – всі бачили як виглядає ситуаційна кімната, де на великих екранах малюються діаграми, а за їх динамікою слідкує персонал.

- ➢ Сприяння інноваціям.
  Моніторинг дозволяє вчасно виявляти нові можливості та впроваджувати інноваційні рішення, що відповідають стратегії організації, а саме оновлювати програмне забезпечення, змінювати складові «заліза», удосконалювати захист.

Зупинимось на принципах збору даних при системному моніторингу, що включають в себе наступні аспекти:

- ➢ Визначення ключових метрик (KPIs)

Визначаємо основні параметри та метрики, які ви хочете моніторити. Це можуть бути продуктивність, швидкість реакції, завантаження системи, обсяги даних тощо.

- Вибір засобів моніторингу
  Обираємо інструменти та програмне забезпечення, які дозволяють збирати, аналізувати та візуалізувати дані. Це можуть бути спеціальні моніторингові системи або власноруч розроблені рішення.
- Налаштування моніторингу
  Конфігуруємо інструменти моніторингу для збору даних з обраних метрик та параметрів. Встановлюємо правила та порогові значення, за якими буде виявлено аномалії.
- Збір та агрегація даних
  Систематично збираємо дані з обраного спектру метрик. Це може включати отримання інформації про ресурси системи, вхідні та вихідні дані, відгуки користувачів тощо.
- Аналіз та інтерпретація даних
  Проводимо аналіз отриманих даних для виявлення аномалій, прогнозування проблем та визначення можливих шляхів вирішення. Порівнюємо зі значеннями KPIs та пороговими значеннями.
- Візуалізація результатів
  Використовуємо графіки, діаграми та інші засоби візуалізації для наглядного представлення результатів моніторингу. Це полегшить розуміння стану системи та дозволить приймати обґрунтовані рішення.
- Звітність та документація
  Ведемо детальну документацію процесу моніторингу, результати аналізу та прийняті рішення. Це стане важливим джерелом інформації для подальших удосконалень та оптимізацій.
- Автоматизація та оптимізація
  Обов’язково передбачаємо можливість автоматизації процесу моніторингу для ефективності та точності збору даних. На основі аналізу результатів вдосконалюємо стратегію моніторингу та параметри аналізу.

3. Основні принципи та цілі системного моніторингу

Сучасні інформаційні технології є невід'ємною складовою діяльності бізнесу та організацій у всіх сферах. У цьому контексті системний моніторинг має критичне значення для забезпечення надійності, ефективності та безпеки цих технологічних екосистем.

Системний моніторинг ґрунтується на кількох ключових принципах, спрямованих на ефективне та ефективне спостереження, аналіз та управління інформаційними системами та їхніми компонентами [21].

Ціль системного моніторингу полягає в забезпеченні надійності, ефективності та безпеки системи шляхом постійного спостереження за її функціонуванням та негайної реакції на будь-які невідповідності або проблеми.

Необхідно згадати таке поняття, як проактивність — уявлення про природу психіки людини. Згідно з уявленнями про проактивність психіки людини, між подразниками, що впливають на людину і її реакцією на ці подразники лежить вільна незалежна воля. Тобто особистіть сама обирає як реагувати на ті чи інші дії. Протилежністю проактивності є реактивність, коли вибір визначається зовнішніми обставинами, стимуляцією.

Чому тут згадано наче б то далеке від предмету моніторингу поняття психоогії? А тому, що воно використовується також і у моніторингу як антипод пасивного спостереження за об’єктами.

Перелічимо основні принципи проактивного моніторингу та реагування на проблеми. Це узагальнені принципи, які справедливі для більшості видів моніторингу:

- Проактивність
  - Системний моніторинг передбачає постійний нагляд за параметрами системи для виявлення можливих проблем чи відхилень ще до того, як вони стануть критичними.
  - Проактивність дозволяє адміністраторам та інженерам уникати серйозних збоїв та реагувати на відхилення задовго до впливу на користувачів чи бізнес-процеси.
- Автоматизоване сповіщення
  - Використання систем автоматичного сповіщення для оперативного повідомлення адміністраторів про виявлені аномалії чи проблеми.
  - Автоматичне сповіщення дозволяє вчасно реагувати та вирішувати проблеми, навіть якщо адміністратор не постійно на зв'язку.
- Прогнозування проблем
  - Використання аналітики та алгоритмів прогнозування для виявлення тенденцій та передбачення можливих проблем на основі зібраних даних.
  - Прогнозування проблем дозволяє приймати запобіжні заходи та уникнути можливих негативних наслідків.

Деякі джерела надають більш розгорнутий перелік принципів системного моніторингу. Він виглядає наступним чином [20]:

- Цільова орієнтованість (Goal-Oriented)
  Моніторинг повинен бути спрямований на досягнення конкретних цілей та задач. Цілі можуть включати покращення продуктивності, забезпечення безпеки, забезпечення надійності, оптимізацію ресурсів тощо. Вся система моніторингу повинна бути адаптована до цих цілей.
- Реалізація метрик та показників (Metrics and Indicators)
  Моніторинг повинен включати визначення та вимірювання ключових метрик та показників, які відображають стан системи. Це може бути продуктивність, завантаження, швидкість відгуку, використання ресурсів та інші важливі параметри.
- Континуїтет та постійність (Continuous Monitoring)

Системний моніторинг повинен бути постійним та безперервним процесом. Це дозволяє вчасно виявляти зміни, аномалії та проблеми у системі, щоб оперативно реагувати та уникнути серйозних наслідків.

- Автоматизація та інтеграція (Automation and Integration)
  Автоматизація збору та аналізу даних є ключовим принципом. Це дозволяє оптимізувати час та ефективність процесу моніторингу. Інтеграція моніторингу з іншими інструментами та системами теж важлива для забезпечення повноцінного управління системою.
- Виявлення та попередження (Detection and Alerting)
  Система моніторингу повинна бути здатна вчасно виявляти аномальні стани, відхилення та проблеми. Крім цього, важливо мати систему сповіщень, яка негайно повідомляє відповідальних осіб про виникнення проблем.
- Скорочення "шуму" (Noise Reduction)
  Моніторинг повинен бути спрямований на виділення суттєвих змін та проблем, уникання зайвого "шуму" в інформації. Це дає можливість концентруватися на дійсно важливих подіях та проблемах.
- Скалабельність та гнучкість (Scalability and Flexibility)
  Системний моніторинг повинен бути побудований таким чином, щоб ефективно працювати як для невеликих, так і для великих систем, а також бути гнучким та адаптованим до змін у вимогах та конфігураціях систем.

Основні цілі системного моніторингу, зокрема підвищення доступності, продуктивності та безпеки систем, узагальнено та наведено у табл. 1.2.

Ці принципи та цілі системного моніторингу допомагають забезпечити стабільність, ефективність та безпеку інформаційних систем у сучасному IT-середовищі.

Таблиця 1.2

Цілі системного моніторингу

Ціль системного моніторингу	**Мета цілі**	**Заходи забезпечення**
Покращення доступності (Ensuring continuous availability)	Забезпечення неперервної доступності систем та послуг для користувачів	Моніторинг функціональності та виявлення точок можливого збою, щоб уникнути перебоїв у роботі
Підвищення продуктивності (Increasing task execution efficiency)	Максимізація ефективності використання ресурсів та оптимізація продуктивності системи	Моніторинг ресурсів, виявлення перевантажень, оптимізація алгоритмів та розподіл ресурсів для максимальної продуктивності
Забезпечення безпеки систем (Ensuring system security)	Захист системи від потенційних загроз та вразливостей	Моніторинг активності, виявлення підозрілих або нестандартних взаємодій, вдосконалення системи безпеки на основі зібраних даних
Оптимізація ресурсів (Optimization of resource utilization)	Раціональне використання ресурсів для максимальної продуктивності при мінімальних витратах	Моніторинг використання ресурсів (процесор, пам'ять, мережа), розподіл завдань, ідентифікація ефективних алгоритмів роботи

4. Види метрик, що вимірюються при системному моніторингу

Метрика у контексті системного моніторингу є стандартизованим способом вимірювання певних характеристик та параметрів системи, щоб оцінити її продуктивність, надійність, ефективність та інші аспекти функціонування. Це ключовий елемент для аналізу та оцінки роботи системи та прийняття обґрунтованих управлінських рішень.

Зрозуміло, що метрика у моніторингу це свого роду лінійка для вимірювання деяких показників чи параметрів. Якщо говорити про види метрик, то "**Технічні метрики**", "**Метрики доступності**" і "**Бізнес метрики**" є основними та ключовими категоріями метрик. Крім них є ще декілька інших важливих видів метрик, які можна врахувати в класифікації [42], [2].

- **Споживання ресурсів.** Оцінює використання ресурсів, таких як електроенергія, батарея, простір для зберігання, та інші.
- **Метрики безпеки.** Визначають рівень безпеки системи, включаючи виявлення загроз, управління доступом та аудиторію.
- **Метрики ефективності.** Вимірюють ефективність процесів та робочих потоків в організації чи системі.
- **Метрики відновлення (Recovery Metrics).** Оцінюють час відновлення послуги або системи після виникнення збоїв.
- **Метрики користувацького досвіду.** Визначають якість та задоволення користувача при використанні системи або сервісу.
- **Метрики навантаження (Load Metrics).** Моніторять обсяг роботи, який система чи сервіс повинен обробити в певний період часу.
- **Метрики вартості (Cost Metrics).** Визначають витрати та ефективність з точки зору фінансового аспекту.

До речі, кожна метрика, у міру своїх характеристик, може одразу входити у кілька різних класів. Як наприклад основні технічні метрики є також базовими метриками у метриках споживання ресурсів.

Технічні метрики - це конкретні вимірювання технічних параметрів чи характеристик в інформаційних технологіях та комп'ютерних системах. Ці метрики використовуються для кількісної оцінки та відстеження різних аспектів функціонування технічних систем з метою покращення їхньої продуктивності, надійності, ефективності та безпеки. Технічні метрики включають в себе вимірювання різних ресурсів (таких як процесор, пам'ять, диск) і параметрів мережі, часу реакції системи, завантаження серверів, швидкості передачі даних тощо. Вони є ключовим інструментом у системному моніторингу та управлінні IT-системами.

При виборі технічних метрик для моніторингу важливе значення відіграють граничні значення технічних метрик, що визначаються для вчасного виявлення аномалій, попередження можливих проблем та ефективного управління

ресурсами системи. Встановлення граничних значень базується на специфікаціях системи, її вимогах та типі діяльності. Загальний підхід до визначення граничних значень для типових технічних метрик наведено у табл. 1.3.

Таблиця 1.3

Типові технічні метрики

Технічна метрика	**Опис метрики**	**Значення метрики**	**Граничні (порогові) значення метрики**
Використання процесора	Визначає, який відсоток обчислювальної потужності процесора використовується	Високе використання процесора може вказувати на перевантаження, що може призвести до зниження продуктивності	**Ultra Low**: Less than 10% CPU usage **Normal**: 10-70% usage **Warning**: 70-90% usage **Critical**: More than 90% utilization
Використання пам'яті	Відображає відсоток доступної оперативної пам'яті, яка використовується системою	Високе використання пам'яті може свідчити про неефективне використання ресурсів або потребу в додатковій пам'яті	**Ultra Low**: Less than 20% memory usage **Normal**: 20-70% usage **Warning**: 70-90% usage **Critical**: More than 90% utilization
Використання дискового простору	Вимірює, який обсяг дискового простору використовується на системному диску	Високе використання може сигналізувати про недостатність місця та необхідність оптимізації або розширення простору	**Ultra Low**: Less than 30% disk space usage. **Normal**: 30-70% usage. **Warning**: 70-90% usage. **Critical**: More than 90% utilization
Пропускна здатність мережі	Вимірює кількість даних, які можуть бути передані через мережу за певний час	Зменшення пропускної здатності може вказувати на проблеми з мережею, що впливають на швидкість передачі даних	**Ultra Low**: More than 80% available bandwidth. **Normal**: 50-80% usage. **Warning**: 30-50% usage. **Critical**: Less than 30% usage
Відповідь сервера	Час, який сервер витрачає на обробку та відповідь на запити	Довгий час відповіді може призвести до низької продуктивності та погіршення враження користувачів	**Ultra low**: Response less than 100ms **Normal**: 100-500 ms **Warning**: 500-1000 ms **Critical**: More than 1000 ms
Кількість запитів до бази даних	Кількість запитів, які виконуються в базі даних за певний час	Збільшення цього показника може вказувати на підвищений обсяг роботи з базою даних та потребу в оптимізації	**Ultra Low**: Less than 100 requests per minute **Normal**: 100-500 requests **Warning**: 500-1000 requests **Critical**: More than 1000 requests
Швидкість передачі даних	Визначає швидкість передачі даних між компонентами системи чи зовнішніми ресурсами	Повільна швидкість передачі може впливати на продуктивність та реакцію системи на користувацькі запити	**Ultra Low**: More than 1 Mbps of free bandwidth **Normal**: 500 kbps - 1 Mbps of used bandwidth **Warning**: 200 Kbps - 500 Kbps of used bandwidth **Critical**: Less than 200 kbps of free bandwidth

Наведені у табл. 1.3 граничні значення технічних метрик є лише прикладом, який базується на «чистій теорії» та аналізі трендів. Тому необхідно відмітити три моменти при встановленні граничних значень метрик

- **Контекст застосування**: Граничні значення слід визначати, враховуючи конкретні вимоги та характеристики системи.
- **Аналіз трендів**: Застосування аналізу трендів та історичних даних для визначення оптимальних граничних значень.
- **Поглиблений моніторинг**: Для деяких систем може бути необхідно використовувати більш детальні граничні значення для конкретних компонентів.

Встановлення граничних значень метрик моніторингу - ітеративний процес, який вимагає постійного відстеження та аналізу, оскільки вони можуть змінюватися в залежності від потреб системи та змін у її навколишньому середовищі.

Метрики використання ЦП необхідно підбирати для кожного серверу індивідуально. Для деяких систем завантаження 91-92% (особливо хмарних серверів) це нормально і дозволяє суттєво економити бюджет на цих серверах. Також є сенс виключати якісь некритичні для функціонування системи об'єкти з моніторингу: DVD-приводи (фізичні або змонтовані), підключені мережні диски, які відслідковуються на своїх серверах, тимчасові диски хмарних серверів.

Було розглянуто основні технічні метрики. Зрозуміло, що вони відіграють ключову роль у системному моніторингу, допомагаючи забезпечити ефективність, надійність та оптимальне використання ресурсів у IT-інфраструктурі.

Зупинимось на такому класі метрик, як **метрики доступності** у відношенні системного моніторингу [21], [20].

Метрики доступності в системному моніторингу визначаються як конкретні вимірювання, які оцінюють доступність та надійність інформаційної системи чи сервісу. У контексті системного моніторингу ці метрики визначають ступінь готовності та працездатності системи для виконання своїх завдань. Додатково, ці метрики дозволяють визначати і враховувати час відмов та час відновлення.

Що ж таке та сама доступнісь? Доступність означає здатність елемента конфігурації або IT-послуги виконувати узгоджену функцію, коли це необхідно. Доступність розраховується за допомогою рівняння:

$$Availability\% = \frac{(Agreed\ Service\ Time - DownTime)}{Agreed\ Service\ Time},$$

де *Agreed Service Time* (Узг*оджений час обслуговування*) – це очікуваний час, коли послуга працюватиме. Якщо ваш рівень обслуговування визначає, що користувачі повинні мати доступ до системи хмарного бекапу з 6:00 ранку до півночі в робочі дні, узгоджений час обслуговування становить 18 годин/1080 хвилин/64 800 секунд на робочий день.

DownTime (*час простою*) –час, протягом якого послуга недоступна протягом узгодженого часу.

Доступність вимірюється як відсоток часу, протягом якого послуга або елемент конфігурації доступні. Він звітує про минуле та оцінює майбутнє

служби. Він говорить, наскільки добре послуга працювала протягом періоду вимірювання.

Показники доступності також оцінюють, наскільки добре послуга працюватиме в майбутньому. У нашому прикладі доступності системи хмарного бекапу середня доступність 99,99% передбачає, що ми можемо очікувати середній час безвідмовної роботи нашої служби 17,9982 годин/1079,892 хвилин/64 793,52 секунд на день. Ми можемо порівняти розраховані дані з обіцяною доступністю, щоб визначити, чи досягаємо ми своїх бізнес-цілей.

Дуже дивно, але класна статистика доступності не завжди перетворюється на хороші результати для клієнтів. Тут може мати місце маркетингово-психологічний термін, що носить назву «ефекту кавуна» (рис. 1.4), коли постачальник послуг досягає мети вимірювання, але не підтримує бажані результати клієнта.

Припустімо, ви постачальник телекомунікаційних послуг із щотижневою доступністю 99,9% (0,1% або 10 хвилин простою на тиждень). Але цей простой у 0,1% відбувається під час інтенсивних подій, таких як звітний день, коли необхідно перелити по каналам велику кількість даних, або під час відеоконференції у момент заключення дуже вигідного контракту. Цільові показники доступності досягнуто, але ваші клієнти незадоволені. Це класичний візерунок кавуна, зелений (добре) зовні, червоний (погано) всередині.

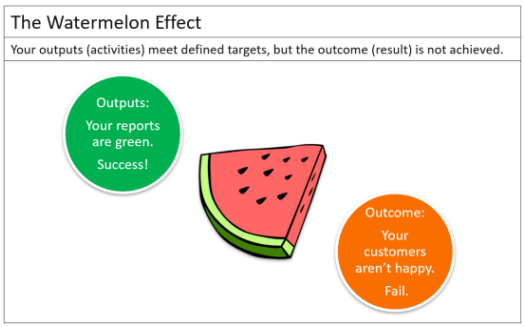


Рис. 1.4. «Ефект кавуна»

Три основних метрики доступності, їх визначення та описи доступності наведені у табл. 1.4.

Ці метрики дозволяють визначити рівень надійності та готовності системи, її доступність для користувачів та взаємозв'язок з продуктивністю.

Зрозуміло, що метрики доступності відіграють роль для споживача лише поки система в цілому зберігає працездатність. Тут можна навести безліч прикладів. Самий яскравий, та життєвий приклад, з яким ми всі регулярно маємо справу – доступність інтернету. Вам, як клієнту, провайдер може розповідати про час відновлення 30 хв, але якщо зв'язок зник у важливий для нас відрізок часу, значення метрик доступності відіграє якусь роль лише для самого провайдера.

Таблиця 1.4

Основні метрики доступності

Назва метрики	Визначення метрики	Опис метрики

Доступність системи (System Availability)	Доступність системи визначає, як часто система знаходиться в стані готовності та працездатності без відмов.	Ця метрика вимірюється у відсотках та вказує на частоту, з якою система може виконувати свої функції без перебоїв.
Час відмови (Downtime)	Час відмови представляє собою період, протягом якого система або сервіс не доступний або не працює належним чином.	Вимірюється в одиницях часу (години, хвилини), ця метрика вказує на тривалість відмов та ступінь впливу на доступність.
Час відновлення (Recovery Time)	Час відновлення вказує, як швидко система може бути відновлена до нормальної робочої діяльності після виникнення відмови.	Ця метрика вимірюється у відсотках або конкретних одиницях часу та вказує на ефективність процесів відновлення.

А тепер розглянемо ще дві метрики (табл. 1.5 та 1.6), а саме:

- MTBF (Mean Time Between Failures);
- MTTR (Mean Time To Recovery)

Вони тісно пов'язані з метриками доступності, але вони представляють інші сторони оцінювання надійності та швидкості відновлення системи.

Таблиця 1.5

Додаткові метрики

Назва метрики	Визначення метрики	Опис метрики
Час безвідмовної роботи системи (MTBF, Mean Time Between Failures) Середній час між відмовами	Час безвідмовної роботи (MTBF) вимірює середню тривалість часу між виникненням двох послідовних відмов системи чи компонента.	MTBF вказує на середню тривалість, протягом якої система очікується, що буде функціональною перед тим, як виникне нова відмова. Вищий показник MTBF вказує на більшу надійність системи.
Час відновлення послуги **(MTTR, Mean Time To Recovery)** Середній час відновлення	Час відновлення послуги (MTTR) вимірює час, необхідний для відновлення системи чи послуги після виникнення відмови.	MTTR оцінює ефективність та швидкість відновлення системи після перебоїв. Менший MTTR свідчить про більшу ефективність відновлення та менший вплив відмови на користувачів.

Ці метрики грають ключову роль у вимірюванні та управлінні доступністю системи, допомагаючи прогнозувати та зменшувати відмови та час відновлення для забезпечення найкращого можливого рівня обслуговування для користувачів (табл. 1.6).

MTBF та MTTR не є прямими включеннями до списку метрик доступності, але вони є критичними компонентами для визначення, наскільки доступна та надійна система. Загальний підхід може виглядати так: високий MTBF сприяє підвищенню доступності, а коротший MTTR допомагає зменшити час відмови та покращити загальну доступність системи.

Таблиця 1.6

Метрики управління доступністю системи

Назва метрики	Відношення до доступності	Пов'язаність зі списком метрик

MTBF Середній час між відмовами	Обернено пропорційний до частоти відмов системи. Високий MTBF вказує на низьку частоту відмов, що, в свою чергу, підвищує загальну доступність системи.	MTBF може розглядатися як компонент доступності, але більше відображає ймовірність безвідмовної роботи протягом певного періоду.
MTTR Середній час відновлення	Прямо пропорційний до швидкості відновлення системи після відмови. Має велике значення для скорочення часу відмови та покращення доступності.	MTTR є важливою частиною оцінки доступності, оскільки відображає, як швидко система може повернутися до нормальної роботи після виникнення проблеми.

Згадаємо таку начеб то зовсім не технічну річ, як бізнес-метрики [46], [30]. Бізнес-метрики – конкретні кількісні та якісні вимірювання, які використовуються для оцінки впливу інформаційних технологій та інших бізнес-процесів на стратегічні та фінансові результати організації. Ці метрики допомагають визначити, наскільки ефективно використовуються ресурси, як вони впливають на задоволення клієнтів, фінансові показники та інші аспекти діяльності компанії. У табл. 1.7 перелічено визначальні риси бізнес-метрик.

Таблиця 1.7

Бізнес метрики та їх риси

Орієнтованість на стратегічні цілі	Бізнес-метрики повинні бути прив'язані до стратегічних цілей організації та допомагати визначити, наскільки успішно вони досягаються.
Кількісний та якісний характер	Метрики можуть бути числовими (наприклад, прибуток, витрати) або якісними (наприклад, задоволення клієнтів).
Взаємозв'язок з бізнес-процесами	Вони повинні відображати результати та ефективність ключових бізнес-процесів організації.
Відстеження змін в часі	Метрики можуть використовуватися для відстеження змін в часі, що дозволяє організації визначати тенденції та аналізувати ефективність за період.
Орієнтованість на кінцевого користувача	Вони часто відображають вплив на клієнтів, які можуть бути замовниками, споживачами, чи іншими зацікавленими сторонами.
Порівняння та бенчмаркінг	Метрики можуть використовуватися для порівняння результатів зі стратегічними цілями, або з результатами подібних організацій (бенчмаркінг).

У табл. 1.8 узагальнено та наведено приклади типових бізнес-метрик, що використовуються для оцінювання ефективності діяльності інформаційних систем.

Таблиця 1.8

Бізнес метрики та їх характеристики

Назва бізнес-метрики	Характеристика	Розрахунок
ROI (Return on Investment)	Вимірює відношення прибутку від вкладеного капіталу або інвестицій до їхньої вартості.	$ROI = \frac{(\text{Прибуток} - \text{Вартість інвестицій})}{\text{Вартість інвестицій}} \times 100\%$
TCO (Total Cost of Ownership)	Визначає загальну вартість володіння та утримання IT-системи протягом всього періоду експлуатації.	$TCO = \text{Вартість придбання} + \text{Вартість експлуатації} + \text{Вартість обслуговування}$
Customer Lifetime Value (CLV)	Визначає прибуток, який очікується від одного клієнта протягом всього його перебування в системі.	$CLV = \text{Середній дохід на клієнта} \times \text{Середній термін обслуговування клієнта}$
Conversion Rate	Визначає відсоток користувачів, які здійснили бажану дію, наприклад, покупку, реєстрацію чи завантаження.	$Conversion\ Rate = \frac{(\text{Кількість бажаних операцій})}{\text{Загальна кількість відвідувачів}} \times 100\%$
Churn Rate	Визначає відсоток користувачів, які припинили використання послуг або продуктів компанії.	$Churn\ Rate = \frac{(\text{Кількість відтоку клієнтів})}{\text{Загальна кількість клієнтів на початку періоду}} \times 100\%$

Ці метрики допомагають бізнес-лідерам розуміти вартість та вплив IT-систем на стратегічні та фінансові аспекти діяльності компанії.

Згадаємо ще дві важливі бізнес-метрики, які допомагають оцінити ефективність IT-інфраструктури та визначити стратегії її вдосконалення. Це ***витрати на технічне обслуговування та відновлення після збоїв*** та ***втрати доходу внаслідок перерв у роботі системи***.

Витрати на технічне обслуговування можуть включати в себе витрати на утримання та обслуговування обладнання, витрати на віддалене адміністрування, зарплати технічного персоналу, витрати на оновлення програмного забезпечення тощо.

Витрати на відновлення після збоїв включають в себе витрати на роботу збійної системи, вартість втраченого часу праці персоналу, витрати на відновлення даних, витрати на відновлення та поновлення служб та інфраструктури.

Втрати доходу внаслідок перерв у роботі системи можна розрахувати, враховуючи втрати прибутку за час перерви у роботі системи. Вони можуть включати втрати від простою обслуговування клієнтів, можливі втрати партнерів, іміджеві втрати та інші фінансові втрати. Дані про тривалість перерви можуть бути зібрані через системний моніторинг, журнали відмов, інформаційні системи для керівництва або від звітів технічного персоналу.

Втрати доходу можна розрахувати, помноживши тривалість перерви на прогнозований або фактичний обсяг прибутку на одиницю часу.

Дані про вплив на клієнтів та імідж можуть бути отримані через відгуки клієнтів, соціальні мережі, або інші канали комунікації з клієнтами.

Перелічені метрики можна відстежувати та розраховувати, використовуючи системи моніторингу, логи подій, аналітичні інструменти та внутрішні дані компанії. Важливо враховувати потреби бізнесу та налаштовувати системний моніторинг для збору необхідних даних для ефективного визначення цих величин.

І на завершення розділу, хотілося б підсумувати, що метрики в системному моніторингу охоплюють різні аспекти системи та її функціонування. Вони дозволяють вимірювати та оцінювати різноманітні параметри, які є ключовими для надійності, ефективності та безпеки систем. Основні типи метрик у системному моніторингу включають:

- Продуктивність. Performance Metrics
  - Час відгуку (Response Time): Час, який система витрачає на обробку запиту та надсилання відповіді.
  - Кількість оброблених запитів на одиницю часу (Requests per Second): Кількість запитів, які система може обробити протягом секунди.
  - Тривалість виконання операцій (Operation Duration): Середній час, який витрачається на виконання конкретних операцій.
- Використання ресурсів. Resource Utilization Metrics
  - Використання центрального процесора (CPU Utilization): Відсоток використання обчислювальної потужності процесора.
  - Використання оперативної пам'яті (Memory Utilization): Відсоток використання доступної оперативної пам'яті.
  - Використання мережного трафіку (Network Traffic Utilization): Кількість переданих даних через мережу в порівнянні з доступною пропускною спроможністю.
- Надійність та доступність. Reliability and Availability Metrics
  - Час доступності (Uptime): Відсоток часу, протягом якого система була доступною для користувачів.
  - Кількість збоїв (Number of Failures): Кількість виявлених проблем або відмов системи.
- Безпека. Security Metrics
  - Кількість виявлених кібератак (Number of Detected Cyber Attacks): Кількість спроб несанкціонованого доступу до системи.
  - Рівень захисту (Security Posture): Оцінка загального рівня захищеності системи та дотримання правил безпеки.
- Масштабованість. Scalability Metrics
  - Розмір та обсяг даних (Data Size and Volume): Кількість та обсяг даних, які система може ефективно обробляти та зберігати.
  - Масштабованість додатку (Application Scalability): Ефективність додатку при збільшенні навантаження та обсягу користувачів.
- Витрати. Cost Metrics

- Витрати на обслуговування (Maintenance Costs): Сума грошей, витрачених на підтримку та обслуговування системи протягом певного періоду.

➢ Інші метрики. Other Metrics

- Скарги користувачів (User Complaints): Кількість та характер скарг від користувачів щодо функціонування системи.

Ці метрики важливі для ефективного моніторингу та управління системою, дозволяючи оцінювати різні аспекти її роботи та вживати заходів для покращення продуктивності, надійності та безпеки.

Висновки до розділу.

У цьому розділі відбулося занурення у світ системного моніторингу, розкриваючи його визначення, роль у сучасних інформаційних технологіях та основні принципи та цілі. Системний моніторинг виявився ключовим елементом сучасного IT-пейзажу, забезпечуючи нагляд та управління найважливішими аспектами інфраструктури та сервісів.

Визначивши системний моніторинг як процес збору та аналізу даних про стан системи, виявилось, що його роль в сучасних IT-технологіях непереоціненна. Цей інструмент дозволяє не лише реагувати на поточні проблеми, а й передбачати їх, щоб уникнути серйозних відмов та оптимізувати ресурси.

Основні принципи та цілі системного моніторингу виявилися спрямованими на забезпечення доступності, надійності та ефективності інфраструктури. Ми розглянули важливість вчасного виявлення аномалій, моніторингу різних аспектів, таких як продуктивність, безпека та використання ресурсів.

Окрему увагу ми приділили видам метрик, які вимірюються при системному моніторингу. Використовуючи технічні метрики, такі як використання процесора, пам'яті, дискового простору та інші, системи можуть отримати повний обсяг інформації про свій стан та продуктивність.

Системний моніторинг - це не тільки інструмент для виявлення проблем, але і стратегічний ресурс для вдосконалення та оптимізації роботи IT-інфраструктури. Правильне впровадження та налагодження системного моніторингу є важливим етапом для забезпечення стабільності та ефективності сучасних інформаційних систем.

Закінчимо розділ так само, як і почали — будь-який системний адміністратор знає, що моніторинг є важливим. Іноді — дуже важливим.

Контрольні запитання

1) До якого виду моніторингу відноситься дода́тковий моніторинг?
2) Що означає абревіатура MTBF?

3) Метрика - це визначення певних характеристик системи, які міняються випадковим чином?
4) Метрика дискового простору вимірює загальний обсяг дискового простору системи?
5) Чи можна віднести температурний моніторинг до типу інфраструктурного?
6) Узагальнені принципи інфраструктурного моніторингу: автоматизоване сповіщення, проактивність та прогнозування проблем
7) На метрики доступності впливає узгоджений час обслуговування?
8) Повільна швидкість передачі може впливати на продуктивність та реакцію системи на користувацькі запити.
9) Повільна швидкість передачі негативно впливає на загальну продуктивність системи.

Розділ 2. Архітектура системного моніторингу

1. Компоненти системи моніторингу: агенти, сервери, бази даних

Архітектура системного моніторингу - це структурована організація компонентів, процесів та взаємодії між ними в системі моніторингу для забезпечення ефективного збору, аналізу та відображення інформації про стан і функціонування інформаційно-технічної системи.

Розглянемо спрощений узагальнений алгоритм моніторингу (рис. 2.1). Відповідно до цієї спрощеної моделі, система моніторингу складається з чотирьох основних компонентів: спостереження, аналізу, дії та зберігання [61], [39].

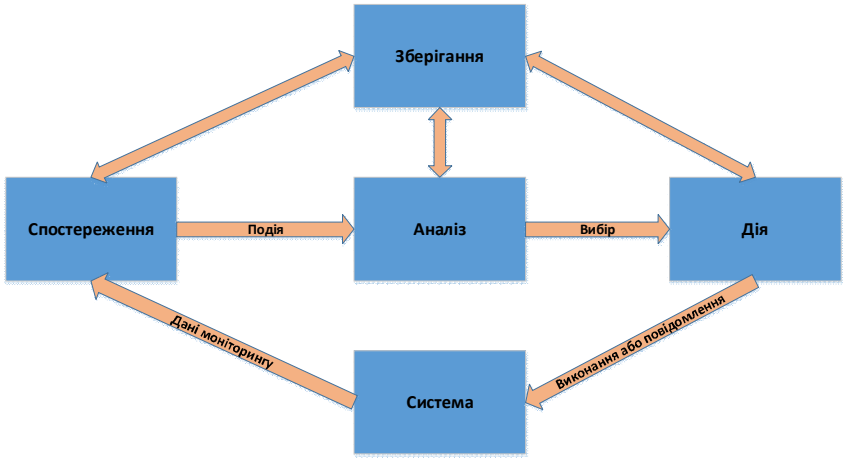


Рис. 2.1. Узагальнений алгоритм моніторингу

Відповідно до цієї моделі, алгоритм моніторингу виконує чотири завдання, які також характеризують основні компоненти алгоритму, як показано на рис. 2.1. Компонент спостереження монітора отримує дані з точки спостереження за системою (або підсистемою) та генерує події моніторингу для подальшого аналізу.

Компонент зберігання даних двобічно взаємодіє з усіма іншими компонентами алгоритму.

Компонент аналізу реалізує політику моніторингу, тобто правила та умови, які необхідно перевірити. Якщо виявлено порушення, моніторинг запускає дії, які можуть варіюватися від простого звітування до примусового виконання певної поведінки в цільовій системі. Прикладами примусового виконання є просте припинення процесу, плавне вимкнення, заходи відновлення або коригуюче втручання.

Як вже згадувалося, всі інші компоненти алгоритму мають доступ до компонента зберігання для виконання різноманітних завдань, наприклад, для резервного копіювання необроблених даних або контрольних журналів, як бази знань для аналізу або для дій з реєстрації.

Спрощений алгоритм розглянуто. Переходимо до архітектури системного моніторингу, яка зазвичай включає в себе наступні ключові компоненти та взаємозв'язки, а саме:

- **Джерела даних.** Це компоненти, які постачають інформацію для моніторингу. Це може включати мережне обладнання, сервери, програмне забезпечення, бази даних тощо. Джерела даних можуть використовувати різні протоколи та інтерфейси для передачі інформації [61], [52].
- **Збір даних.** Цей компонент відповідає за збір даних від джерел та їхнє передавання до системи моніторингу. Включає в себе агенти, сенсори, протоколи (наприклад, SNMP, NetFlow) та інші механізми для отримання інформації [53], [52], [41].
- **Основна система моніторингу.** Це ядро системи, де збираються, аналізуються та зберігаються дані. Включає в себе базу даних для зберігання історії подій, движок правил для виявлення аномалій, систему управління для організації процесів моніторингу [39], [52].
- **Візуалізація та звітність.** Компонент, який відповідає за представлення інформації користувачеві. Включає в себе інтерфейс користувача, графіки, діаграми, засоби відображення статусів та інші елементи для зручного сприйняття даних.
- **Система сповіщень та реагування.** Цей компонент відповідає за генерацію сповіщень при виявленні проблем або аномалій. Також може включати механізми автоматичного реагування або взаємодії з іншими системами для вирішення проблем.
- **Безпека.** Компонент, що забезпечує захист системи моніторингу від несанкціонованого доступу, шифрування даних та інші заходи для збереження конфіденційності та цілісності інформації.
- **Адміністративний інтерфейс.** Інтерфейс, який дозволяє адміністраторам налаштовувати та керувати системою моніторингу, встановлювати правила, конфігурувати засоби візуалізації та інші параметри.

Компоненти архітектури взаємодіють між собою для створення комплексної системи моніторингу, яка забезпечує адміністраторам інформацію про стан та продуктивність інформаційно-технічної інфраструктури та допомагає вчасно реагувати на можливі проблеми.

На рис. 2.2 показана розширена діаграма взаємозв’язків компонентів системного моніторингу, що включає роль людини в цій системі. Трошки конкретики про наведені у діаграмі компоненти.

- **Джерела даних (Network Devices, Servers, etc.)**: надають інформацію для моніторингу і є ключовими компонентами для збору інформації з інформаційно-технічної інфраструктури. Дані від цих джерел допомагають адміністраторам моніторити та аналізувати стан систем, вчасно виявляти проблеми, планувати розширення та оптимізацію мережі.

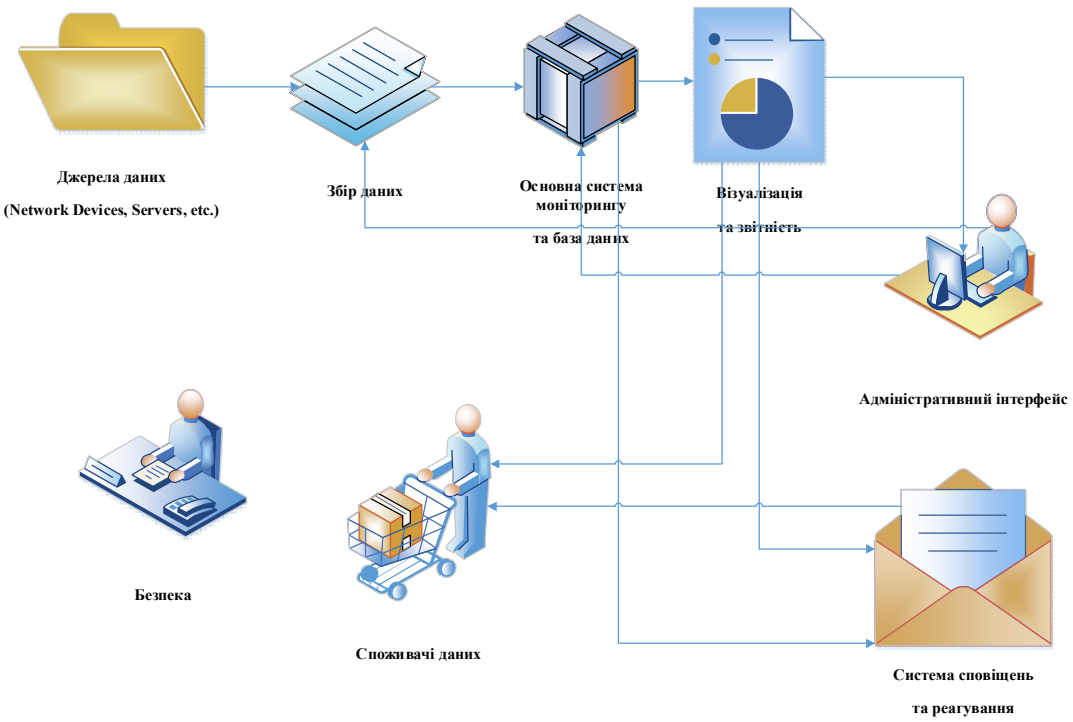


Рис. 2.2. Розширена діаграма, яка відображає взаємозв'язки між компонентами архітектури системного моніторингу

У реальному середовищі системного моніторинга ці компоненти можуть виглядати наступним чином:

✓ **Network Devices (Мережне обладнання)**. Може включати маршрутизатори, комутатори, файрволи, мережні перетворювачі та інше. Для моніторингу такого обладнання можна використовувати протоколи, такі як SNMP (Simple Network Management Protocol), які надають інформацію про стан мережного обладнання, трафік, завантаження та інші метрики. Мережному моніторингу буде присвячена одна з наступних лекцій даного курсу [18], [39], [41].

✓ **Servers (Сервери)**. Фізичні або віртуальні сервери, які надають різні сервіси та додатки. Для моніторингу серверів використовують агенти (програми, що встановлюються на сервері) або взаємодіють з різними протоколами, такими як SNMP або WMI (Windows Management Instrumentation) [61], [39], [53].

✓ **Other Devices (Інші пристрої)**. Включає різноманітне обладнання, таке як датчики температури, вологості, UPS (забезпечення електроживлення обладнання у випадку відсутності чи аномалій в основному живленні), принтери та інше. Для моніторингу цих пристроїв можуть використовуватися різні протоколи або спеціалізовані програмні рішення.

✓ **Applications (Додатки)**. Це програмне забезпечення, яке надає певні сервіси або функціональність. Для моніторингу додатків можуть використовуватися різні підходи, такі як взаємодія з API, використання агентів або моніторинг системних ресурсів, пов'язаних з додатком. Як приклад перевірка бази даних додатка або перевірка логів. У самому простому випадку – перевірка стану процесу, що ми навчимося гарно робити під час виконання лабораторних робіт.

✓ **Security Devices (Засоби безпеки)**. Включають файрволи, системи виявлення та запобігання вторгненням (IDS/IPS), антивіруси та інші засоби

безпеки. Для цих пристроїв використовуються протоколи та інтерфейси моніторингу безпеки. Тут треба згадати такий клас систем, споріднених до систем системного моніторингу (вибачте за тавтологію) як системи виявлення атак (вторгнень) (англ. Intrusion Detection System, IDS) Одразу приходять на думку такі гарні абревіатури як Suricata, Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), Snort. Але зараз ми не будемо зупинятися на цій цікавій та перспективній тематиці, бо вона лежить трошки не у темі лекції.

- **Збір даних (Data Collection)**: збирає дані (хоча це і тавтологія, пробачте) від джерел та передає їх до основної системи моніторингу. У структурі системного моніторингу - це процес отримання інформації про стан і функціонування різних компонентів системи. Цей процес має на меті збирати дані з джерел, таких як мережне обладнання, сервери, додатки, або інші пристрої, які підлягають моніторингу, і передавати ці дані для подальшого аналізу та візуалізації в системі моніторингу. Збір даних є важливим етапом в системі моніторингу, оскільки від якості та точності цього процесу залежить ефективність моніторингу та здатність системи реагувати на зміни стану. Після збору даних вони можуть бути використані для визначення трендів, виявлення аномалій, та взагалі оцінки продуктивності та доступності системи. Основні етапи та компоненти процесу збору даних:

✓ **Визначення джерел даних**. Визначення того, з яких пристроїв чи джерел системи буде здійснюватися збір інформації. Це може включати мережне обладнання (роутери, комутатори), сервери, додатки, датчики, принтери, WEB-сайти та інше.

✓ **Вибір методу збору**. Вибір підходів та методів для отримання інформації з обраного джерела. Наприклад, для мережного обладнання часто використовують протоколи, такі як SNMP (Simple Network Management Protocol) [54], а для серверів - агенти моніторингу чи інші протоколи (наприклад, WMI для Windows).

✓ **Конфігурація агентів чи засобів збору**. Якщо використовуються агенти, їх потрібно налаштувати для збору конкретних метрик або даних. Налаштування включає в себе вибір параметрів моніторингу, інтервалів збору, та інші налаштування залежно від потреб системи.

✓ **Передача даних**. Передача зібраних даних до центральної системи моніторингу. Це може включати передачу за допомогою різних протоколів, API-запитів, або інших механізмів взаємодії.

✓ **Агрегація та зберігання**. Агрегація та зберігання зібраних даних в базі даних для подальшого аналізу, архівування та створення звітів.

- **Основна система моніторингу (Monitoring Core System) та база даних (DataBase)**: компонент відповідає за приймання, обробку та аналіз зібраних даних в реальному чи поблизькому до реального часу. Основні функції цього компоненту включають [61], [52]:

✓ **Прийом та обробка даних**. Отримання інформації від джерел даних (збірники, агенти, інші джерела) та її подальша обробка. Це включає валідацію даних, конвертацію в однорідний формат, та виявлення можливих аномалій чи проблем.

✓ **Зберігання даних**. Дані, отримані від різних джерел, зберігаються в базі даних для подальшого використання. Звичайно, зберігання включає архівацію історичних даних, що дозволяє вести аналіз та виявляти тренди протягом тривалого періоду.

✓ **Аналіз та моніторинг**. Виконання аналізу даних для виявлення аномалій, визначення трендів та подальший моніторинг. Аналітичні засоби можуть включати в себе різні алгоритми, правила, та методи для ідентифікації проблем чи аномалій у системі.

✓ **Генерація звітів та візуалізація**. Виведення результатів аналізу у вигляді графіків, діаграм, звітів та інших візуальних засобів для зручного сприйняття та розуміння стану системи.

✓ **База даних** в контексті системного моніторингу є сховищем для зберігання різноманітної інформації, отриманої від джерел даних. Основні аспекти її функціонування включають:

❖ **Зберігання інформації**. База даних зберігає отримані дані про стан системи, її компоненти, метрики та інші параметри. Це дозволяє ведення історії подій та зберігання інформації для подальшого аналізу.

❖ **Архівація та запити**. Забезпечення можливості архівації історичних даних та виконання запитів для вибірки необхідної інформації. Це дозволяє адміністраторам та системам аналізувати та використовувати дані для різноманітних потреб.

❖ **Оптимізація доступу**. Оптимізація швидкості доступу до даних для ефективного використання системи моніторингу, особливо коли обсяг даних є великим.

❖ **Безпека даних**. Забезпечення безпеки зберігання та доступу до даних. Це включає в себе встановлення прав доступу, шифрування та інші заходи для забезпечення конфіденційності та цілісності даних.

Основна система моніторингу та база даних тісно співпрацюють, забезпечуючи збір, аналіз, та зберігання інформації для ефективного моніторингу та управління інфраструктурою. Звичайно, що бази даних це велика окрема тема. Варто відмітити, що більшість систем моніторингу працюють з «легкими», безкоштовними базами типу MySQL або чогось свого, пропрієтарного.

- **Візуалізація та звітність**: є важливим елементом архітектури системного моніторингу, оскільки надає адміністраторам та користувачам зручний та зрозумілий інтерфейс для сприйняття та аналізу зібраних даних. Основні функції цього компонента включають:

✓ **Графіки та діаграми**. Надають можливість побудови графіків та діаграм для візуалізації трендів, змін та стану різних параметрів системи. Це допомагає адміністраторам швидко розпізнати зміни та проблеми.

✓ **Дашборди**. Система дозволяє створювати та налаштовувати дашборди, де можна об'єднувати різні елементи візуалізації для зручного спостереження за ключовими метриками та подіями.

✓ **Звіти та аналітика**. Надають можливість створення звітів та аналітичних звітів на основі зібраних даних. Адміністратори можуть генерувати різноманітні звіти для аналізу ефективності системи та її компонентів.

✓ **Реальний час**. Забезпечує можливість відстежування даних в реальному часі. Це дозволяє оперативно реагувати на події та аномалії, що відбуваються в системі.

✓ **Фільтрація та групування**. Забезпечують можливість фільтрації та групування даних для зручного відображення конкретної інформації. Це особливо корисно в системах з великою кількістю компонентів та метрик.

✓ **Повідомлення та сповіщення**. Інтеграція системи візуалізації з системою сповіщень для оперативного інформування адміністраторів про важливі події чи аномалії.

✓ **Користувацькі налаштування**. Надають можливість адміністраторам налаштовувати інтерфейс з урахуванням їхніх власних потреб та візуальних уподобань.

Компонент візуалізації та звітності важливий для забезпечення адміністраторів інструментами для ефективного моніторингу та управління системою. Він допомагає перетворити складні дані в зрозумілу форму, полегшуючи процес прийняття рішень та реагування на події в реальному часі.

- **Система сповіщень та реагування**: виявляє аномалії та сповіщає адміністраторів та є критично важливою для забезпечення надійності та продуктивності системи. Цей компонент дозволяє адміністраторам оперативно реагувати на події та забезпечує швидке виправлення проблем для забезпечення неперервності роботи інфраструктури. Основні аспекти цього компонента включають:

✓ **Виявлення аномалій**. Система використовує аналітичні алгоритми та правила для виявлення аномальних станів або несподіваних змін в параметрах системи. Це може включати в себе порогові значення, шаблони поведінки та інші методи виявлення незвичайних ситуацій.

✓ **Сповіщення**. При виявленні аномалії система надсилає сповіщення адміністраторам або відповідальним особам. Сповіщення може бути реалізоване через різні канали, такі як електронна пошта, SMS, месенджери чи інші засоби зв'язку.

✓ **Конфігурування правил**. Адміністратори можуть налаштовувати правила сповіщень відповідно до потреб системи та їхніх власних пріоритетів. Це включає визначення умов для сповіщень, методів доставки, та інших параметрів.

✓ **Система пріоритетів**. Врахування різних рівнів важливості для різних типів сповіщень. Деякі події можуть вимагати негайної реакції, тоді як інші можуть бути позначені як менш важливі.

✓ **Інтеграція з іншими системами**. Система Сповіщень може інтегруватися з іншими системами, такими як системи керування інцидентами чи автоматизовані засоби реагування на проблеми.

✓ **Автоматичне реагування**. Можливість налаштовувати автоматичне реагування на деякі типи подій чи аномалій без прямого втручання адміністраторів. Це може включати автоматичне відновлення послуг, виключення проблемних компонентів, то що.

✓ **Журналізація та відстеження**. Запис подій та реакцій в журналах для подальшого аналізу та аудиту. Це дозволяє адміністраторам вивчати історію подій та визначати ефективність заходів реагування.

- **Споживачі даних** - це сутності або системи, які використовують зібрані дані для виконання різноманітних завдань. У якості споживачів даних у системах моніторингу можуть виступати автоматизовані системи, програми, скрипти, а також самі адміністратори, чи навіть звичайні користувачі системи моніторингу

(звичайно, якщо Ви як системний адміністратор налаштуєте їм певний доступ) які взаємодіють з системою через адміністративний інтерфейс. Основні ролі споживачів даних включають:

✓ **Автоматизовані системи та інтеграція**. Системи моніторингу часто інтегруються з іншими автоматизованими інфраструктурними системами. Наприклад, інтеграція з системами управління інцидентами або іншими інструментами для автоматизованого реагування на проблеми.

✓ **Споживачі API**. *Абревіатура "API" в даному випадку означає "Інтерфейс програмування застосунків" (англ. Application Programming Interface).* Споживачі можуть використовувати API для отримання даних з системи моніторингу. Це може бути корисно для створення власних звітів, інтеграції з іншими інструментами або створення власних автоматизованих сценаріїв.

✓ **Системи моніторингу продуктивності**. Споживачі даних можуть бути спрямовані на оцінку продуктивності системи моніторингу. Це може включати в себе аналіз завантаження, швидкості відгуку та інших параметрів для забезпечення ефективності системи.

✓ **Адміністраторські застосунки та інтерфейс**. Самі адміністратори, використовуючи адміністративний інтерфейс, є споживачами даних. Вони взаємодіють з системою для налаштування параметрів, виявлення аномалій, аналізу даних та прийняття рішень.

✓ **Генерування звітів та аналітика**. Споживачі можуть використовувати дані для створення різноманітних звітів, аналітичних висновків та статистичних даних для внутрішніх чи зовнішніх потреб.

✓ **Аналіз трендів та прогнозування**. Системи аналізу даних можуть використовувати зібрані дані для виявлення трендів, прогнозування можливих проблем, та прийняття стратегічних рішень.

✓ **Інтерфейс кінцевого користувача**. У деяких випадках, кінцеві користувачі або клієнти можуть бути споживачами даних, якщо система моніторингу надає можливість перегляду певних метрик або стану для зовнішніх зацікавлених сторін.

- **Адміністративний інтерфейс**: дозволяє адміністраторам налаштовувати систему та приймати рішення на основі отриманих даних і є ключовим для забезпечення адміністраторам зручного та ефективного інструмента для конфігурації системи, прийняття рішень та взаємодії з іншими компонентами, налаштовувати параметри системи моніторингу відповідно до потреб та вести контроль над всіма аспектами моніторингу та управління інфраструктурою. Основні характеристики цього компонента включають:

✓ **Керування конфігурацією**. Надає можливість налаштовувати параметри системи моніторингу, включаючи джерела даних, правила моніторингу, інтервали збору, та інші параметри. Це може включати в себе інтерфейс для додавання, вилучення чи зміни джерел даних.

✓ **Налаштування правил сповіщень**. Дозволяє адміністраторам налаштовувати правила та умови, за якими генеруються сповіщення. Це включає в себе визначення порогових значень, типів повідомлень та адреси для сповіщень.

✓ **Управління користувачами та доступом**. Забезпечує інтерфейс для керування обліковими записами користувачів, а також налаштування рівнів доступу та привілеїв для різних користувачів системи моніторингу.

✓ **Дашборди та візуалізація**. Дозволяє адміністраторам створювати та налаштовувати дашборди для зручного відображення ключових метрик та інформації. Це дозволяє швидко отримувати огляд стану системи.

✓ **Керування звітами та аналітикою**. Надає інтерфейс для створення та конфігурації звітів, аналітичних висновків та інших інструментів для детального аналізу даних.

✓ **Моніторинг продуктивності системи**. Надає засоби для відстеження та оцінки продуктивності самої системи моніторингу. Це може включати в себе метрики щодо завантаження системи, швидкості відгуку та інші параметри.

✓ **Інтеграція з іншими системами**. Забезпечує можливість інтеграції з іншими інфраструктурними системами та інструментами для взаємодії та обміну даними.

✓ **Журналізація та аудит**. Запис подій та дій адміністраторів для подальшого аудиту та вивчення історії змін.

- **Безпека**: Забезпечує захист системи від несанкціонованого доступу та збереження конфіденційності даних та відіграє ключову роль у захисті важливих інформаційних ресурсів та забезпеченні конфіденційності, цілісності та доступності даних в системі моніторингу. Основні аспекти цього компонента включають:

✓ **Аутентифікація та авторизація.**

Аутентифікація: Визначення і перевірка ідентичності користувачів або систем, що намагаються **отримати** доступ до системи моніторингу. Це може включати в себе використання паролів, біометричних даних чи інших методів.

Авторизація: Визначення прав доступу та привілеїв для користувачів або систем, які вже автентифіковані. Забезпечення того, що користувачі мають лише **той** рівень доступу, який їм необхідний.

✓ **Шифрування даних**. Застосування шифрування для захисту даних, які передаються між компонентами системи моніторингу. Це може включати шифрування даних в покоїному стані, а також шифрування під час передачі.

✓ **Моніторинг та аудит безпеки**. Забезпечення системи функціоналом моніторингу та аудиту подій для виявлення можливих загроз, несанкціонованих спроб доступу та аномалій в системі.

✓ **Захист від вторгнень**. Впровадження заходів для запобігання вторгненням в систему моніторингу. Це може включати в себе виявлення та блокування невірних спроб доступу, антивірусний контроль та інші заходи.

✓ **Фізична безпека**. Захист фізичних ресурсів, на яких базується система моніторингу, таких як серверні приміщення, центральні сервери та комунікаційні канали.

✓ **Методи аутентифікації та захисту API**. Забезпечення безпеки взаємодії з іншими системами через API, включаючи використання токенів, ключів доступу та інших методів аутентифікації та захисту.

✓ **Захист інфраструктури**. Забезпечення безпеки компонентів інфраструктури, таких як бази даних, сервери та мережні пристрої, для запобігання несанкціонованого доступу та атак.

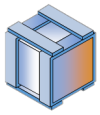
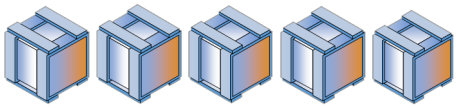
✓ **Резервне копіювання та відновлення**. Реалізація стратегій резервного копіювання та відновлення даних для забезпечення можливості відновлення системи після інцидентів або атак.

2. Розгорнуті та розподілені архітектури моніторингу

Виділяють розгорнуту та розподілену архітектури моніторингу. У табл. 2.1 наведено порівняльну цих архітектур, зокрема їхні ключові компоненти та сфери застосування.

Таблиця 2.1

Порівняння розгорнутої та розподіленої архітектур моніторингу

Розгорнута архітектура моніторингу [45], [39]. 	**Розподілена Архітектура Моніторингу [45], [23].** 
Характеристики: • **Централізована**. Всі компоненти, включаючи агенти та сервери моніторингу, розташовані в одному центральному місці. • **Одна центральна база даних**. Використання однієї центральної бази даних для збереження всіх зібраних даних. • **Масштабування за допомогою збільшення ресурсів**. Масштабування системи досягається за рахунок збільшення ресурсів на центральному сервері.	**Характеристики:** • **Розподілена інфраструктура**. Компоненти системи розташовані на різних фізичних чи віртуальних серверах, можливо, розташованих географічно чи топографічно в різних місцях. • **Багатоцентрові бази даних**. Використання багатоцентрових баз даних, можливо, розташованих в різних регіонах або областях.
Ключові Компоненти: • **Агенти**. Інстальовані на кожній системі, яку потрібно моніторити. • **Центральний Сервер**. Обробляє та зберігає дані, відправлені агентами. • **Центральна база даних**. Зберігає всі дані з моніторингу.	**Ключові Компоненти:** • **Агенти**. Розподілені на системах у різних місцях. • **Локальні сервери**. Сервери моніторингу, які розташовані ближче до джерел даних. • **Глобальні сервери**. Центральні сервери, які об'єднують та агрегують дані з різних локальних серверів. • **Багатоцентрові бази даних**. Централізовані та локальні бази даних для ефективного зберігання та обробки інформації.
Застосування: Для невеликих або середніх мереж, де централізованого підходу вистачає для обробки обсягу даних.	**Застосування:** • **Глобальні організації**. Для компаній або організацій з розподіленим географічним розташуванням джерел даних. • **Великі IT-інфраструктури**. Для великих комплексних систем, де обсяг даних великий і розподілений.

Різниця між розгорнутою та розподіленою архітектурою полягає в наступному:

- **Централізований контроль vs. децентралізований контроль**. Розгорнута архітектура забезпечує централізований контроль, тоді як розподілена архітектура реалізує децентралізований підхід з розподіленими серверами та базами даних.
- **Масштабування**. Розгорнута архітектура масштабується головним чином за рахунок збільшення ресурсів на центральному сервері. Розподілена архітектура може масштабуватися горизонтально за рахунок додавання нових серверів у різних локаціях.
- **Вартість трафіку**. Розгорнута архітектура може мати більшу кількість трафіку між агентами та центральним сервером. У розподіленій архітектурі частина трафіку може бути оброблена локально перед відсиланням на центральний рівень.

Вибір між розгорнутою та розподіленою архітектурою залежить від:

- **Масштабу проекту**. Для невеликих та середніх проектів розгорнута архітектура може бути більш простою. Розподілена архітектура підходить для великих та розгалужених інфраструктур.
- **Географічного розташування**. Якщо джерела даних розташовані географічно розподілено, розподілена архітектура може бути ефективнішою.
- **Вимог безпеки та збереження доступності**. Якщо важливість безпеки та доступності велика, розподілена архітектура може забезпечити більший рівень стійкості до відмов та атак.

Вибір між розгорнутою та розподіленою архітектурою системи моніторингу не є остаточним або незворотним рішенням. Такий вибір може переглядатися та змінюватися в залежності від вимог до системи, навантаження, обсягів даних та інших факторів, що визначають її ефективність. Значною мірою це залежить від архітектури ключових складових системи моніторингу, зокрема Monitoring Core System та бази даних (Database), які забезпечують обробку, зберігання та аналіз зібраної інформації.

Monitoring Core System та Database є центральними компонентами сучасних систем моніторингу, і саме їхні можливості визначають потенціал системи щодо масштабування, підвищення продуктивності та переходу до розподіленої архітектури. Вибір відповідних технологій для цих компонентів дозволяє гнучко адаптувати систему до змінних умов експлуатації та потреб організації.

У табл. 2.2 наведено приклади таких «ядер систем моніторингу» та стандартних (з коробки) баз даних, які найчастіше використовуються разом із ними.

Наведені приклади показують різноманітність систем моніторингу та їхніх баз даних, що може включати в себе різні комбінації для різних потреб та вимог.

Таким чином, обираючи між цими архітектурними підходами та типами компонентів системи моніторингу, слід враховувати конкретні вимоги проекту, його розмір та географічні особливості [18], [35], [53].

Таблиця 2.2

Приклади «ядер систем моніторингу» та стандартних (з коробки) баз даних

Core System	База даних	Короткий опис	Взаємодія з БД
Prometheus	**Prometheus Database**	Відкрита система моніторингу та оповіщення, яка призначена для збору та відображення метрик з систем та додатків	Використовує власну базу даних для зберігання зібраних метрик. База даних Prometheus оптимізована для швидкого читання та запитів на дані метрик.
Grafana	**InfluxDB**	Платформа візуалізації та аналізу даних, яка працює поверх інших систем моніторингу, таких як InfluxDB	InfluxDB представляє собою спеціалізовану систему для зберігання та обробки часових рядів даних. Ця база даних часових рядів є ідеальним вибором для зберігання та отримання метричних даних, які використовуються Grafana для подальшої візуалізації та аналізу.
Nagios	**MySQL**	Система моніторингу, яка використовується для виявлення та реагування на проблеми в інфраструктурі.	У деяких випадках Nagios може використовувати MySQL або інші бази даних для зберігання конфігураційних даних та даних стану системи.
Zabbix	**PostgreSQL**	Комплексна система моніторингу, яка надає різні функції, включаючи збір, аналіз та візуалізацію даних.	Зазвичай Zabbix використовує PostgreSQL для зберігання конфігураційних даних, статистики та журналів.
Dynatrace	**Cassandra**	Рішення для моніторингу та управління додатками, яке використовує штучний інтелект для аналізу продуктивності.	Cassandra може використовуватися для зберігання метричних даних, які вивчає Dynatrace для аналізу
MS System Center Operations Manager (SCOM)	**Microsoft SQL Server**	Система моніторингу та управління для операційних систем, інфраструктури та додатків у середовищах Windows. SCOM дозволяє виявляти та реагувати на проблеми у реальному часі, а також здійснювати аналіз продуктивності.	SCOM використовує базу даних для зберігання конфігураційних даних, статусів моніторингу та інших важливих інформаційних показників. Зазвичай, для цього використовується Microsoft SQL Server.

3. Протоколи зв'язку між компонентами системи моніторингу

Система моніторингу є ключовим елементом для забезпечення стабільності, ефективності та безпеки інформаційних технологій та бізнес-процесів і одразу зрозуміло, що ефективний зв'язок між компонентами цієї системи є дуже важливим, оскільки він впливає на наступні аспекти функціонування та результативності:

- **Збір та передача даних**. Ефективний зв'язок дозволяє надійно та швидко збирати дані з агентів та інших джерел моніторингу і відповідно - затримки або втрати при передачі можуть призвести до неповної чи неправильної інформації, ускладнюючи процес моніторингу та реагування на проблеми.
- **Швидкість реагування на аномалії**. Зв'язок визначає швидкість обміну інформацією про виникнення аномалій чи проблем в системі, а затримки у виявленні або сповіщенні про аномалії можуть погіршити реакцію на критичні ситуації та збільшити час простою системи.
- **Масштабування та ресурсоємність**. Забезпечення ефективного зв'язку необхідно для масштабування системи та розподілених ресурсів. Неправильно спроектований зв'язок може призвести до перевантаження мережі та серверів, обмежуючи можливість системи моніторингу працювати в розподіленому середовищі.
- **Безпека та конфіденційність**. Зв'язок повинен бути забезпечений для захисту передачі конфіденційної інформації та управлінських команд. Небезпечне, або незахищене з'єднання може стати джерелом загроз безпеці та джерелом витоку конфіденційної інформації.
- **Інтеграція та сумісність**. Інтеграція з іншими системами та сумісність з різними пристроями та платформами вимагають ефективного зв'язку. Несумісність може призвести до втрати функціональності та неефективного використання ресурсів.
- **Автоматизація та динамічна керованість**. Ефективний зв'язок дозволяє автоматизувати процеси моніторингу та приймати динамічні управлінські рішення. Поганий зв'язок може ускладнити впровадження автоматизації та обмежити можливості динамічного керування.
- **Легкість управління та налаштування**. Зв'язок повинен бути простим у налаштуванні та управлінні для забезпечення ефективного функціонування системи моніторингу. Складний зв'язок може збільшити час та ресурси, необхідні для управління та підтримки системи.

Ефективність зв’язку між компонентами будь якої системи забезпечується каналами зв’язку та ефективним вибором протоколу.

В системі моніторингу ефективність зв’язку забезпечує надійність, продуктивність та безпеку обміну даними між компонентами. Окрім вибору протоколу та каналів зв'язку, існує кілька інших факторів (табл. 2.3), які суттєво впливають на ефективність системного моніторингу.

Таблиця 2.3

Фактори впливу на ефективність систем моніторингу

Фактор зв’язку	Опис фактору	Вплив фактору
Пропускна здатність каналів зв'язку	Пропускна здатність визначає, скільки даних може бути передано через канал зв'язку за одиницю часу	Велика пропускна здатність дозволяє передавати більше даних, що особливо важливо в сучасних великих мережах
Затримка та латентність	Затримка показує час, необхідний для передачі сигналу від відправника до отримувача. Латентність визначає загальний час затримки для обробки та передачі даних	Низька затримка та латентність дозволяють швидше реагувати на події та отримувати актуальну інформацію
Протоколи та кодеки компресії	Використання ефективних протоколів та кодеків для стиснення даних може значно зменшити обсяг передаваних інформаційних пакетів	Зменшення обсягу даних покращує швидкість передачі та економить мережний трафік
Безпека та шифрування	Використання безпечних протоколів та шифрування даних забезпечує конфіденційність та інтегрітет інформації	Забезпечення безпеки мережі та даних є важливим для запобігання несанкціонованому доступу та збереження конфіденційності
Резервування та відновлення	Використання механізмів резервування та відновлення дозволяє побудувати стійку до відмов мережну інфраструктуру	Забезпечення надійності та стійкості мережі шляхом автоматичного відновлення та переключення на резервні маршрути
Балансування навантаження	Розподіл навантаження між різними каналами зв'язку або серверами для оптимізації використання ресурсів	Балансування навантаження допомагає уникнути перевантаження та забезпечити рівномірне використання ресурсів
Масштабо-ваність	Можливість системи збільшити свою пропускну здатність та функціональність при збільшенні об'єму оброблюваних даних	Масштабованість гарантує ефективність зв'язку при зростанні обсягу даних та обчислень

Протоколи в системі моніторингу відіграють важливу роль у забезпеченні ефективного обміну даними та взаємодії між різними компонентами, такими як агенти, сервери та інші елементи інфраструктури. Вони створюють стандартизовані правила та процедури, які дозволяють різним частинам системи ефективно спілкуватися та виконувати свої функції. Розуміння та використання різних протоколів в системі моніторингу є важливим елементом для забезпечення ефективного та надійного обміну даними, що визначає успішність функціонування системи та якість зібраних інформаційних даних. Роль протоколів у забезпеченні обміну даними та взаємодії є такою:

- **Стандартизація комунікації**. Протоколи визначають стандартні правила та формати для обміну даними між агентами та серверами, що забезпечує єдність у способі взаємодії, незалежно від типу пристрою чи платформи.

Стандартизована комунікація спрощує інтеграцію різних компонентів та полегшує підтримку системи.

- **Забезпечення надійності та цілісності даних**. Протоколи визначають механізми для перевірки та підтвердження доставки даних, а також їх цілісності під час передачі. Це гарантує, що дані доставлені правильно та не пошкоджені, а забезпечення надійності даних дозволяє уникнути втрати інформації та запевняє точність результатів моніторингу.
- **Управління безпекою**. Протоколи включають механізми шифрування та аутентифікації, щоб забезпечити безпеку обміну даними між агентами та серверами. Управляння безпекою дозволяє захистити конфіденційні інформаційні ресурси. Захист від несанкціонованого доступу та збереження конфіденційності даних ускладнюється без ефективних протоколів безпеки.
- **Оптимізація трафіку та ресурсів**. Протоколи оптимізують передачу даних, управляючи розміром пакетів, використовуючи стиснення та мінімізуючи надлишкові запити. Ефективний обмін даними сприяє економії мережного трафіку та ресурсів, забезпечуючи оптимальну продуктивність системи.
- **Підтримка розподілених систем**. Протоколи дозволяють взаємодіяти між розподіленими компонентами системи моніторингу через мережу, що сприяє розширенню системи на великі відстані, взаємодії з різними джерелами та використанню розподілених обчислювальних ресурсів.
- **Підтримка різних платформ та протоколів**. Протоколи можуть бути розроблені для роботи на різних платформах та підтримувати різні версії протоколів для сумісності з різними версіями програмного забезпечення, що забезпечує сумісність з різними системами та версіями, гарантує гнучкість та розширюваність системи моніторингу.
- **Спрощення управління та налаштування**. Протоколи, які мають чіткі та прості правила, спрощують управління та налаштуванням системи моніторингу, що зменшує час та ресурси, необхідні для внесення змін у конфігурацію та підтримки системи.

Таким чином, протоколи зв'язку грають ключову роль у взаємодії між різними компонентами системи моніторингу, забезпечуючи ефективний обмін даними та забезпечуючи спільну мову для взаємодії між агентами, серверами та іншими елементами інфраструктури.

У табл. 2.4. показано кілька основних протоколів, які широко використовуються в системах моніторингу.

Таблиця 2.4

Протоколи систем моніторингу

Протокол	**Короткий опис**	**Найпоширеніші компоненти**
SNMP (Simple Network Management Protocol)	SNMP є одним з найпоширеніших протоколів для моніторингу та управління мережними пристроями.	Агенти SNMP розташовані на кожному пристрої, який моніториться, та

	Використовується для збору та передачі інформації про стан пристроїв, таких як роутери, комутатори та сервери.	центральному сервері, який здійснює опитування.
Зазвичай при використанні SNMP присутні керовані та керівні системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керівній системі. По суті SNMP агенти передають управлінську інформацію на керівні системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість процесів, що працюють» тощо). Керівна система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM. Керівні системи можуть також відправляти конфігураційні оновлення або контрольні запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі. Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керівної Інформації' (англ. Management Information Bases (MIBs)).		
HTTP/HTTPS (Hypertext Transfer Protocol/Secure)	Протокол HTTP/HTTPS використовується для взаємодії між веб-інтерфейсами адміністратора та серверами моніторингу. Забезпечує передачу даних через веб-браузер та можливість віддаленого керування	Клієнтські браузери та веб-сервери
TCP/IP (Transmission Control Protocol/Internet Protocol)	Основний стек протоколів для мережного зв'язку. Використовується для передачі даних між різними компонентами системи моніторингу, зокрема між агентами та центральними серверами	Усі компоненти системи моніторингу, що спілкуються через мережу.
MQTT (Message Queuing Telemetry Transport)	Легкий та ефективний протокол для передачі повідомлень між пристроями, особливо корисний для IoT-систем та моніторингу великої кількості розподілених пристроїв.	Брокер MQTT, який обробляє та маршрутизує повідомлення, та клієнти, такі як агенти чи інші системні компоненти.

<table>
<tr><td colspan="3">Протокол орієнтований на простоту використання, невисоке навантаження на канали зв'язку, роботу за умов постійної втрати зв'язку, легке вбудовування у систему. Основне призначення - робота з телеметрією від різних датчиків та пристроїв.
Використання шаблону передплатника дозволяє пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або зумовлені, зокрема, протокол не вводить обмежень на формат даних, що передаються.</td></tr>
<tr><td>WMI (Windows Management Instrumentation)</td><td>Використовується для моніторингу та управління ресурсами в операційних системах Windows. Дозволяє отримувати інформацію про систему та виконувати операції над нею</td><td>Агенти WMI на системах Windows та центральний сервер.</td></tr>
<tr><td colspan="3">WMI складається з набору розширень до моделі драйвера Windows, яка забезпечує інтерфейс операційної системи, через який інструментальні компоненти надають інформацію та сповіщення. WMI — це реалізація Microsoft стандартів Web -Based Enterprise Management (WBEM) та Common Information Model (CIM) від Distributed Management Task Force (DMTF).
WMI дозволяє мовам сценаріїв (таким як VBScript або Windows PowerShell) керувати персональними комп’ютерами та серверами Microsoft Windows як локально, так і віддалено.</td></tr>
<tr><td>Syslog</td><td>Використовується для запису подій та повідомлень в системному журналі, що є важливим для моніторингу та діагностики проблем в реальному часі.</td><td>Агенти, які генерують системні події, та сервери, які обробляють та аналізують ці події.</td></tr>
<tr><td>RMI (Remote Method Invocation)</td><td>Використовується для виклику методів об'єктів, що виконуються на віддалених серверах. Застосовується в розподілених системах моніторингу для взаємодії між компонентами.</td><td>Клієнтські та серверні компоненти, які викликають та виконують методи.</td></tr>
<tr><td colspan="3">RMI є механізмом виклику методів об'єктів, які виконуються на віддалених серверах у мережному середовищі. По суті, він дозволяє викликати методи об'єктів, що знаходяться на іншому вузлі мережі, так само, як це робиться для локальних об'єктів. RMI є ключовою технологією для розробки розподілених систем у Java.</td></tr>
</table>

Хоча RMI є вбудованим механізмом у Java, це не означає, що він обмежений тільки Java. Механізм RMI може бути використаний іншими мовами програмування, такими як Python, за допомогою відповідних бібліотек. Існують такі різновиди RMI: • **Java RMI**: Це офіційний механізм реалізації RMI у Java. Використовується для виклику методів об'єктів, що знаходяться на віддалених серверах у мережі. • **CORBA RMI (IIOP RMI)**: CORBA (Common Object Request Broker Architecture) також має свій варіант RMI, який використовує протокол IIOP (Internet Inter-ORB Protocol) для взаємодії між об'єктами. **Jini RMI**: Jini є технологією, що реалізує динамічне підключення та взаємодію розподілених компонентів. Вона також використовує RMI для забезпечення комунікації між цими компонентами.		
Custom APIs	Деякі системи моніторингу можуть використовувати власні API для обміну даними між своїми компонентами або з іншими системами	Залежить від конкретної реалізації, але може включати агенти, сервери та інші компоненти системи.

Кожен з цих протоколів має свої унікальні характеристики та застосування, і вибір залежить від конкретних вимог та контексту системи моніторингу.

Висновки до розділу

У розділі детально розглянуто архітектуру системного моніторингу, розкрито ключові компоненти, які визначають ефективність та функціональність цієї системи.

Спочатку виконано розгорнутий аналіз основних компонентів: агенти, сервери та бази даних. Агенти, які встановлені на об'єктах моніторингу, відповідають за збір та передачу даних, сервери обробляють і аналізують отриману інформацію, а бази даних забезпечують зберігання та доступ до історично-статистичних даних.

Далі, розглянуто дві основні архітектурні моделі - розгорнуту та розподілену. Розгорнута архітектура підходить для менших мереж, де компоненти моніторингу зосереджені на одному сервері. З іншого боку, розподілена архітектура дозволяє розміщувати компоненти на різних вузлах, що дозволяє масштабувати та забезпечувати більшу надійність.

Окрема увага приділена протоколам зв'язку, які виступають ключовим елементом ефективного обміну даними між компонентами системи моніторингу. Відмічено важливість стандартизації комунікації, надійності та безпеки в обраному протоколі.

Ефективність системного моніторингу залежить від правильного підбору архітектурного підходу, належного вибору протоколів та компетентної реалізації компонентів. Лише узгоджена робота всіх елементів забезпечить надійне та ефективне функціонування системи моніторингу.

Контрольні запитання

1) Протокол SNMP використовується для передачі аудіо- та відеоданих між мережними пристроями.

2) Агенти SNMP розташовані лише на центральному сервері, який здійснює моніторинг мережі.

3) Узагальнений алгоритм моніторингу полягає у збереженні даних та двобічній взаємодії безпосередньо з об'єктом моніторингу.

4) Головна мета функціонування елементу архітектури CM Data Collection – збір даних з джерел (мережне обладнання, сервери, додатки, інші пристрої) які підлягають моніторингу, і передавати ці дані для подальшого аналізу та візуалізації в системі моніторингу.

5) Значення абревіатури WMI

6) Розгорнута архітектура моніторингу орієнтована на великі та розгалужені інфраструктури?

7) Розгорнута архітектура моніторингу передбачає децентралізований підхід до архітектури моніторингу?

8) Латентність зв’язку – це часовий інтервал, від моменту відправлення сигналу чи запиту до моменту приходу відповіді чи реакції на нього, включаючи час затримки в обробці та передачі даних?

9) Латентність зв’язку визначає загальний час затримки для обробки та передачі даних?

10) Чи можлива трансформація розгорнутої архітектури моніторингу у розподілену, або розподіленої у розгорнуту в залежності від географії елементів системи моніторингу?

11) Використання багатоцентрових баз даних характерно для розподіленої архітектури моніторингу?

Розділ 3. Вступ до мережного моніторингу

1. Визначення мережного моніторингу

Моніторинг мережі — це використання системи, яка постійно відстежує комп'ютерну мережу на наявність повільних або несправних компонентів і сповіщає адміністратора мережі (через електронну пошту , SMS або інші сигнали тривоги) у разі збоїв або інших проблем. Моніторинг мережі є частиною керування мережею [29].

Ще одне визначення

Моніторинг мережі — це практика моніторингу та аналізу продуктивності та працездатності комп'ютерної мережі, щоб переконатися, що вона працює ефективно та результативно. Він передбачає безперервне вимірювання та оцінку різних параметрів мережі для виявлення та вирішення проблем, оптимізації мережних ресурсів і підтримки високого рівня якості обслуговування [27].

У сучасну цифрову епоху моніторинг мережі наряду з системним та безпековим моніторингом став одним з ключових аспектів управління IT-інфраструктурою. Здатність відстежувати продуктивність мережі та керувати нею має важливе значення для забезпечення ефективної та ефективної роботи мереж. Моніторинг продуктивності мережі може служити керівною силою для IT-фахівців, допомагаючи їм орієнтуватися в складнощах сучасної мережі. Середовищ [28].

В рамках мережного моніторингу використовуються спеціальні засоби та технології для збору, аналізу та візуалізації даних, що дозволяє адміністраторам мережі отримувати комплексну інформацію про стан мережі, виявляти аномалії, вирішувати технічні проблеми та підтримувати високий рівень її ефективності [33].

Ці систематичні дії направлені на виявлення повільних або несправних мережних компонентів, таких як перевантажені сервери, збій чи зависання елементу мережі, несправні маршрутизатори, несправні комутатори, інші проблемні пристрої. У разі відмови мережі або подібного відключення система моніторингу мережі попереджає адміністратора мережі (**NA**). Моніторинг мережі - це підмножина управління мережею [12].

Моніторинг мережі відрізняється від традиційного системного моніторингу тим, що продуктивність контролюється з точки зору кінцевого користувача та вимірюється між двома точками в мережі. Наприклад:

- Продуктивність між користувачем, який працює в офісі, і додатком, який він використовує в дата-центрі компанії
- Продуктивність між двома офісами в мережі
- Продуктивність між головним офісом та Інтернетом
- Продуктивність між вашими користувачами та хмарою

Ключові аспекти мережного моніторингу не відрізняються від інших видів моніторингу та включають [16]:

- **Збір даних.** Мережний моніторинг включає в себе збір різноманітних даних, таких як пакети, логи подій, використання ресурсів, стан обладнання та інші метрики, які характеризують стан мережі.
- **Аналіз та інтерпретація.** Отримані дані аналізуються для виявлення аномалій, прогнозування проблем та оцінки ефективності мережі. Це може включати в себе виявлення вузьких місць, проблем з безпекою, атак або інших аномалій.
- **Візуалізація.** Використовуються графіки, діаграми та інші засоби візуалізації для наглядного представлення стану мережі. Це допомагає швидше розуміти поточний стан та реагувати на проблеми.
- **Сповіщення та реагування.** Моніторингові системи можуть автоматично генерувати сповіщення або реагувати на виявлені проблеми. Це може включати в себе автоматичне відновлення роботи, відправлення повідомлень адміністраторам або запуск процедур безпеки.
- **Документація.** Ведення документації про мережний моніторинг є важливим для аналізу та вдосконалення системи. Це включає в себе записи про виявлені проблеми, вжиті заходи та результати аналізу.

2. Пояснення основних понять та термінів

Не зайвим буде згадати деякі основні терміни та поняття, що є важливими для розуміння теми мережного моніторингу:

- **Мережа (Network).** Сукупність з'єднаних пристроїв чи систем, які можуть обмінюватися даними між собою. Мережі можуть бути локальними (**LAN**), глобальними (**GAN**), або іншими за масштабом, такими як мережі широкого доступу (**WAN**). Локальні мережі (**LAN**) охоплюють обмежену територію, таку як один будинок, підприємство чи установа. Мережі широкого доступу (**WAN**) включають в себе більші території, такі як регіон, країну чи навіть світ, і зазвичай використовують технології, такі як Інтернет, для з'єднання віддалених місць. Адміністратор мережі (**NA**) відповідає за конфігурацію, управління та підтримку мережного обладнання та програмного забезпечення.

- **Моніторинг (Monitoring)**. Це систематичний контроль та відстеження подій, стану чи активності для отримання інформації та виявлення можливих проблем.
- **Мережний моніторинг (Network Monitoring** [50]**)**. Процес відстеження та аналізу стану мережі, її компонентів та трафіку для забезпечення ефективності, безпеки та доступності.
- **Системи моніторингу мережі (NMS).** Network Monitoring System - програмні засоби, призначені для моніторингу та аналізу мережного трафіку. Вони забезпечують видимість у реальному часі продуктивності мережі та подій безпеки.
- **Трафік (Traffic)**. Об'єм даних, які пересилаються між пристроями у мережі. Трафік може бути виміряний в бітах, байтах або інших одиницях.

- **Пропускна спроможність (Bandwidth)**. Максимальний обсяг даних, який може бути переданий через мережу протягом певного часу.
- **Пакет (Packet)**. Невеликий блок даних, який пересилається через мережу. Пакети містять інформацію про джерело, призначення та самі дані.
- **Протокол (Protocol)**. Стандартні правила та формати для обміну даними між пристроями у мережі. Наприклад, TCP/IP - основний протокол Інтернету.
- **Адреса (Address)**. Унікальний ідентифікатор для пристрою чи служби в мережі. IP-адреси та MAC-адреси є прикладами.
- **Пінг (Ping)**. Команда для перевірки доступності пристрою у мережі шляхом відправлення короткого пакету та очікування відповіді.
- **SNMP (Simple Network Management Protocol)**. Протокол для управління та моніторингу мережних пристроїв, або як звучить перше визначення цього протоколу - Simple Network Management Protocol — простий протокол керування мережею [49]
- **Аномалія (Anomaly)**. Неочікувана чи відхилення від звичайного стану. У випадку мережного моніторингу аномалія показників чи параметрів може вказувати на проблеми в мережі.
- **Firewall**. Програмно-апаратний засіб, що контролює та фільтрує мережний трафік для забезпечення безпеки.

Ці терміни становлять основу для розуміння мережного моніторингу та його ключових аспектів.

3. Роль мережного моніторингу у сучасних IT системах

Якась дивна ситуація при вивченні моніторингу – як тільки починаєш розповідати про роль якоїсь складової, вона виявляється або дуже важливою, або головною ☺

Роль мережного моніторингу у сучасних технологічних системах є необхідним елементом для управління сучасними технологічними системами, особливо з урахуванням зростання обсягів даних, складності мереж та загроз безпеки. Головні аспекти цієї ролі мережного моніторингу [43]:

- **Стабільність та продуктивність.**
  - ✓ **Виявлення проблем.** Мережний моніторинг дозволяє виявляти аномалії та вказівники несправностей в роботі мережі [26].
  - ✓ **Прогнозування перевантажень**. Аналіз трафіку допомагає уникнути перевантажень, планувати ресурси та підтримувати стабільність.
- **Безпека та виявлення загроз.**
  - ✓ **Виявлення інцидентів**. Мережний моніторинг допомагає виявляти атаки, несанкціонований доступ та інші потенційно небезпечні активності.
  - ✓ **Моніторинг заходів безпеки**. Спостереження за ефективністю заходів безпеки та швидке реагування на події безпеки.
- **Відділення проблем та швидке відновлення:**
  - ✓ **Ізоляція проблем.** Мережний моніторинг дозволяє швидко ідентифікувати місце та природу проблем.
  - ✓ **Відновлення після відмов**. Автоматизовані системи моніторингу допомагають відновлювати роботу мережі після виникнення неполадок.
- **Оптимізація ресурсів:**
  - ✓ **Моніторинг використання ресурсів.** Слідкування за використанням обладнання та ресурсів допомагає оптимізувати роботу мережі.
  - ✓ **Планування розширень**. Аналіз трафіку допомагає у плануванні майбутніх розширень та апгрейдів.
- **Відповідність та аналітика:**
  - ✓ **Збір даних для відповідності**. Моніторинг забезпечує збір та архівацію даних для відповідності стандартам та регулятивам.
  - ✓ **Аналіз даних.** Використання аналітики для отримання важливих інсайтів щодо ефективності та стану мережі.

4. Оптимізація роботи мережі

Загальні поняття про стабільність та швидкість мережі необхідні для розуміння та ефективного управління мережною інфраструктурою, що забезпечує стійку та швидку передачу даних [27].

- **Стабільність мережі** вказує на її здатність залишатися в робочому стані без непередбачених відмов чи значних коливань у роботі. Стабільність залежить від надійності обладнання, налаштувань конфігурації, ефективності маршрутизації та відсутності атак або неполадок.
- **Швидкість мережі** вказує на кількість даних, які можуть бути передані через мережу протягом певного часу. Швидкість вимірюється в бітах на секунду (bps), кілобітах на секунду (Kbps), мегабітах на секунду (Mbps), гігабітах на секунду (Gbps) і т.д. На швидкість мережі впливають:
  - ✓ пропускна здатність каналів
  - ✓ якість з'єднань
  - ✓ обсяг трафіку
  - ✓ ефективність мережного обладнання.

Зовсім нещодавно (16.11.2023 р.), співробітниками Датського технічного університету встановлено новий світовий рекорд швидкості передачі інформації через оптичне волокно з використанням одного передавача.

Раніше рекорд належав Інституту технологій з Німеччини, де була досягнута швидкість в 11 терабіт в секунду. Група співробітників по фотонним технологій з Технічного університету Данії значно перевершили своїх колег домігшись швидкості в 43 терабіта за секунду.

Ця швидкість дозволить виконати копіювання гігабайтного файлу за 0,2 мс, а копіювання терабайта займе до 0,18 секунди. У коментарях датських фахівців кажуть, що подібні дослідження дозволять прийти до значно більш високих швидкостей, які необхідні і актуальні в зв'язку з ростом обсягів переданого трафіку, розвитком Wi-Fi і стільникових мереж.

Дослідники з Астонського університету у Великій Британії у березні 2024 року досягли швидкості у 33 терабіт в секунду використовуючи у досліді одинарний одномодовий стандартний оптоволоконний кабель.

У червні 2024 року з’явилась інформація, що команда дослідників з Національного інституту інформаційних і комунікаційних технологій (NICT) у Японії використала інноваційну технологію для передачі даних на швидкості у 44 терабіта за секунду. Вони змогли досягти такого результату завдяки вдосконаленню багатомодових волокон і нових методів мультиплексування.

- **Взаємозв'язок між стабільністю та швидкістю** дуже тісний. Забезпечення стабільності часто пов'язано з оптимізацією швидкості та навпаки. Висока швидкість може призвести до нестабільності, якщо інфраструктура не готова впоратися з великим обсягом трафіку. Іншими словами взаємний вплив швидкості та стабільності мережі можна пояснити наступними трьома тезами:
  - ✓ **Швидкість і стабільність**. Висока швидкість мережі дозволяє швидко передавати дані між пристроями та користувачами, що покращує продуктивність та ефективність роботи. Проте, якщо інфраструктура мережі не готова впоратися з великим обсягом трафіку, це може призвести до перевантаження, втрати пакетів даних та зниження якості обслуговування.
  - ✓ **Стабільність забезпечується інфраструктурою**. Щоб забезпечити стабільність мережі, потрібно мати належно налаштовані маршрутизатори, комутатори, файрволи та інші пристрої, які розподіляють навантаження та контролюють трафік. Також потрібно використовувати механізми керування трафіком, які можуть регулювати швидкість передачі даних та захищати мережу від перевантаження.
  - ✓ **Оптимізація для підвищення продуктивності.** Про загальні підходи у оптимізації швидкості мережі (CDN, кешування контенту та використання високопродуктивних пристроїв маршрутизації та комутації) ми поговоримо трошки пізніше.
- **Фактори впливу**. На стабільність та швидкість впливає кілька факторів:
  - ✓ **Пропускна здатність**. Максимальний обсяг даних, який може бути переданий через мережу протягом певного часу. Одиниці вимірювання ті ж, що у у швидкості мережі (біт/с, Кбіт/с, Мбіт/с). Критерії встановлення пропускної здатності це
    - **Тип мережі.** Варіюється залежно від типу мережі, наприклад, Ethernet, Wi-Fi, або мобільного зв'язку.
    - **Протокол**. Варіюється залежно від протоколу, який використовується для передачі даних, наприклад, TCP, UDP, або HTTP.
    - **Апаратне забезпечення**. Залежить від апаратного забезпечення, яке використовується, наприклад, мережних карт, маршрутизаторів, або комутаторів.

    Існує кілька способів вимірювання пропускної здатності мережі:
    - Тестування швидкості: Ви можете використовувати онлайн-сервіси або програмне забезпечення для тестування швидкості вашого інтернет-з'єднання.

- Моніторинг мережі: Ви можете використовувати програмне забезпечення для моніторингу мережі, щоб відстежувати пропускну здатність мережі протягом певного часу.
- Розрахунок: Ви можете розрахувати пропускну здатність мережі, знаючи ширину смуги пропускання та співвідношення сигнал/шум.
- Важливо зазначити, що пропускна здатність мережі може бути динамічною і може змінюватися залежно від різних факторів, таких як завантаження мережі, наявність перешкод, або оновлення програмного забезпечення.

✓ **Затримка та джиттер**. Затримка (Latency) це час, який займає передача даних від джерела до приймача через мережу. Затримка включає час, необхідний для проходження сигналу через маршрутизатори, комутатори, кабелі та інші мережні пристрої. Зазвичай затримка вимірюється у мілісекундах (мс) або мікросекундах (мкс).

Джиттер (Jitter) - це непередбачувані зміни в затримці передачі даних через мережу. Джиттер може виникати через зміну навантаження на мережу, коливання швидкості передачі даних або інші фактори, які впливають на стабільність мережного з'єднання. Джиттер також вимірюється в мілісекундах (мс) або мікросекундах (мкс).

Загальною метрикою, яка використовується для вимірювання і оцінки затримки та джиттеру, може бути середня затримка та середній джиттер. Ці метрики дають уявлення про типові часи затримки та коливання в мережі.

Важливо зазначити, що низька затримка та джиттер допомагають забезпечити стабільний та швидкий зв'язок в мережі, особливо для додатків, які вимагають реального часу, таких як відеоконференції або онлайн-ігри.

✓ **Втрати пакетів**. Кількість пакетів, які не досягли свого призначення через мережу.

- **Важливість балансу.**

✓ **Оптимізація**. Забезпечення оптимального балансу між стабільністю та швидкістю для задоволення потреб користувачів та бізнес-процесів.

✓ **Керування ресурсами**. Ефективне використання ресурсів, таких як пропускна здатність та енергія, для забезпечення стійкості та продуктивності.

5. Забезпечення стабільності та швидкодії

Вплив стабільності та швидкодії на користувачів та бізнес-процеси є невід'ємною частиною стратегії інфраструктурного управління (табл.

3.1) та визначає успіх організації в сучасному конкурентному середовищі [47].

Таблиця 3.1

Користувачі та бізнес-процеси

Користувачі	Бізнес-процеси
• **Задоволення користувачів**. ✓ **Стабільність**. Стабільна мережа забезпечує безперебійні з'єднання, що призводить до задоволення користувачів. Вони можуть впевнено використовувати сервіси без перебоїв. ✓ **Швидкість.** Швидка передача даних дозволяє користувачам отримувати доступ до ресурсів та послуг без затримок, покращуючи їхню взаємодію з інтернетом та іншими сервісами.	• **Неперервність операцій**. ✓ **Стабільність**. Для бізнесу критично важливо мати стабільну мережу, що дозволяє неперервно проводити операції та обслуговувати клієнтів. ✓ **Швидкість**. Швидка передача даних сприяє ефективності бізнес-процесів, зменшуючи час на виконання завдань та обробку інформації.
• **Продуктивність та ефективність.** ✓ **Стабільність**. Відсутність відмов та перебоїв забезпечує неперервну продуктивність користувачів, що є ключовим для бізнес-процесів. ✓ **Швидкість**. Швидка передача даних підвищує продуктивність та забезпечує ефективність в роботі з ресурсами та інструментами.	• **Конкурентоспроможність.** ✓ **Стабільність**. Організації з надійною мережею здатні уникати втрат часу та ресурсів, що робить їх конкурентоспроможними на ринку. ✓ **Швидкість**. Швидкі бізнес-процеси дозволяють реагувати на зміни ринку швидше, що може бути важливим для успіху підприємства.
• **Відгуки та лояльність**. ✓ **Стабільність.** Стабільна робота мережі призводить до позитивних відгуків і підтримує лояльність користувачів. ✓ **Швидкість.** Швидке обслуговування сприяє задоволенню користувачів, що впливає на їхню готовність залишатися в системі.	• **Безпека даних.** ✓ **Стабільність**. Стабільна мережа важлива для забезпечення безпеки даних та уникнення доступу до них несанкціонованими особами. ✓ **Швидкість**. Швидка передача даних може підвищити ефективність заходів забезпечення безпеки та захисту інформації.

6. Стратегії оптимізації для забезпечення стабільності та швидкодії мережі

У табл. 3.2 узагальнено основні стратегії оптимізації, спрямовані на забезпечення стабільності та підвищення швидкодії мережі, із зазначенням їх призначення та відповідних стратегічних дій.

Таблиця 3.2

Основні стратегії оптимізації

Категорія	Призначення	Стратегічні дії
Керування пропускною здатністю	Полягає в розподілі та контролі доступу до мережної пропускної здатності з метою оптимізації трафіку та покращення швидкодії.	✓ **Пріоритетизація трафіку**. Встановлення пріоритетів для різних типів трафіку (наприклад, відео, голос, дані), щоб важливі дані мали вищий пріоритет та гарантований доступ до пропускної здатності. ✓ **Мережні контролери пропускної здатності**. Використання спеціалізованих пристроїв та алгоритмів для управління пропускною здатністю та виявлення проблем.

Пріоритетизація трафіку може бути особливо важливою в ситуаціях, коли мережа перевантажена або коли важливі додатки чи послуги потребують стабільної пропускної здатності. Ось деякі методи пріоритетизації трафіку:

- **Quality of Service (QoS)** - це механізм, який дозволяє встановлювати пріоритети для різних видів трафіку на основі визначених параметрів, таких як тип даних (голос, відео, дані), протокол чи джерело/призначення. Наприклад, голосовий трафік може мати вищий пріоритет над трафіком відеоконференцій, а відео - над загальними даними.
- **Traffic Shaping (Формування трафіку)** використовується для керування пропускною здатністю шляхом обмеження швидкості передачі даних для певних типів трафіку. Наприклад, швидкість завантаження/відвантаження може бути обмежена для файлового трафіку, щоб забезпечити достатню пропускну здатність для інших важливих додатків, таких як голос або відео.
- **Packet Prioritization (Пріоритетизація пакетів)** полягає у встановленні пріоритету для індивідуальних пакетів даних у мережі. Це може бути досягнуто за допомогою маркування пакетів заздалегідь визначеними мітками (тегами), які вказують рівень пріоритету. Марковані пакети обробляються з вищим пріоритетом, що забезпечує їм перевагу при передачі через мережу.
- **Traffic Classification (Класифікація трафіку)** визначає типи трафіку на основі його характеристик, таких як порт призначення, протокол або структура даних. Після класифікації трафіку можна встановити правила для керування пропускною здатністю відповідно до потреб кожного типу трафіку [33].

Щодо мережних контролерів пропускної здатності (Network Traffic Controllers) - це пристрої або програмне забезпечення, що використовуються для управління трафіком у мережі з метою оптимізації пропускної здатності та виявлення проблем.

Ось деякі аспекти використання спеціалізованих пристроїв та алгоритмів управління пропускною здатністю:

- **Quality of Service (QoS) Policies**. Мережні контролери можуть застосовувати QoS політики для встановлення пріоритетів та керування трафіком на основі параметрів, таких як тип даних, джерело/призначення, протокол тощо. Наприклад, вони можуть надавати вищий пріоритет для голосового трафіку порівняно з відеоабо файловим трафіком.
- **Traffic Shaping and Policing**. Контролери можуть використовувати методи формування трафіку та контролю для керування швидкістю передачі даних. Це

дозволяє регулювати трафік для забезпечення оптимального використання пропускної здатності мережі та уникнення перевантаження.

- **Deep Packet Inspection (DPI)**. Деякі контролери можуть використовувати DPI для аналізу вмісту пакетів даних і виявлення різних типів трафіку або патернів. Це дозволяє виявляти аномальний трафік, шкідливі програми або атаки на мережу.
- **Load Balancing**. Контролери також можуть використовуватися для розподілу трафіку між різними мережними шляхами або серверами з метою збалансування навантаження і забезпечення оптимальної швидкості та доступності.
- **Traffic Monitoring and Analysis**. Контролери також можуть здійснювати моніторинг трафіку та аналізувати дані про його потік для виявлення аномалій, виявлення проблем у мережі або ідентифікації джерела проблем.

Network Traffic Controllers можуть бути реалізовані як спеціалізовані апаратні пристрої, так і програмне забезпечення, що запускається на загальнопризначених серверах або віртуальних машинах. Ці можливості додатково конфігуруються та налаштовуються залежно від потреб конкретної мережі або організації.

Щодо виробників, які займаються випуском та підтримкою Network Traffic Controllers, ось декілька провідних компаній:

- **Cisco Systems**
- **Juniper Networks**
- **Huawei**
- **F5 Networks**
  Palo Alto Networks

Використання технологій кешування	Використовуються для збереження копій часто використовуваних даних близько до користувачів, зменшуючи час доступу та навантаження на мережу.	✓ **Локальне кешування**. Зберігання копій даних на локальних пристроях чи серверах, що дозволяє зменшити час завантаження даних з віддалених джерел. ✓ **Використання CDN (Content Delivery Network)**. Розподілення копій контенту на серверах, розташованих у різних регіонах, для забезпечення швидкого доступу до контенту.

Локальне кешування впливає на стабільність та швидкодію мережі, надаючи такі переваги:

- **Зменшення трафіку**, яке забезпечується тим, що повторні запити до контенту можуть бути оброблені локально, без необхідності витрачати пропускну здатність мережі для запитів цього контенту з віддалених серверів. Це дозволяє зменшити загальний обсяг трафіку в мережі та робить її більш ефективною.
- **Покращена швидкість доступу**. Оскільки контент знаходиться ближче до кінцевого користувача, час доступу до цього контенту зменшується. Кешовані дані можуть бути доставлені швидше, оскільки вони вже знаходяться на локальному пристрої або на сервері, розташованому ближче до користувача.
- **Зменшення завантаження віддалених серверів**. Локальне кешування дозволяє розподіляти навантаження між локальними та віддаленими серверами. Це може зменшити навантаження на віддалені сервери, що дозволяє їм краще керувати запитами та забезпечити стабільнішу роботу.
- **Покращення доступності контенту при відсутності зв'язку з мережею**. Локальне кешування дозволяє користувачам отримувати доступ до контенту, навіть якщо вони втратили зв'язок з мережею. Це може бути особливо корисним для веб-сайтів або додатків, які надають важливу інформацію, доступ до якої може бути критичним навіть в умовах обмеженого зв'язку з Інтернетом.

Коротко щодо технології **Content Delivery Network.** CDN складається з мережі серверів, розташованих у різних частинах світу. Ці сервери можуть належати різним компаніям, які спеціалізуються на наданні послуг CDN. Ось кілька провідних компаній, які надають послуги CDN:

Akamai Technologies має одну з найбільших мереж CDN у світі. Вони надають послуги CDN для різних клієнтів, включаючи великі компанії, які потребують швидкої доставки контенту до своїх користувачів.

Cloudflare також пропонує широкий спектр послуг CDN, включаючи захист від DDoS-атак та інші додаткові функції безпеки.

Amazon Web Services (AWS) CloudFront. Amazon, має власну службу CDN під назвою CloudFront, яка інтегрується з іншими послугами AWS та надає швидку доставку контенту для веб-сайтів та додатків.

Google Cloud CDN. Google також пропонує свою службу CDN, яка інтегрується з іншими послугами Google Cloud і забезпечує швидку доставку контенту для користувачів.

Ці компанії розгортають свої сервери CDN у різних місцях світу, використовуючи глобальні мережі дата-центрів. Вони також відповідають за адміністрування та підтримку цих серверів, включаючи їхню конфігурацію, моніторинг, оновлення та захист від атак. Необхідно відмітити основні переваги використання CDN:

- **Розподілення навантаження.** розподіляє запити на контент між різними серверами, розташованими у різних географічних регіонах. Це дозволяє зменшити навантаження на один сервер та забезпечити більш стабільну роботу мережі, оскільки навантаження рівномірно розподіляється.
- **Кешування контенту.** Зберігає копії веб-сайтів та іншого контенту на своїх серверах. Коли користувач робить запит до веб-сайту, CDN може надати контент з сервера, який знаходиться найближче до користувача географічно. Це дозволяє зменшити час завантаження сторінок та покращити швидкість відгуку веб-сайту.
- **Зменшення відстані.** Оскільки CDN має сервери в різних частинах світу, вона допомагає скоротити фізичну відстань між сервером та кінцевим користувачем. Це зменшує затримку (латентність) при передачі даних і покращує швидкість завантаження контенту.
- **Захист від DDoS-атак.** CDN має захист від DDoS-атак, який допомагає уникнути перевантаження серверів та зберегти стабільність мережі у разі масштабних атак.

Оптимізація маршрутизації	Передбачає вибір та налаштування найкращих шляхів для передачі даних від відправника до отримувача. Забезпечує найкращий шлях для передачі даних, зменшуючи затримки та забезпечуючи стабільність шляху.	✓ **Ефективні протоколи маршрутизації**. Використання протоколів маршрутизації, таких як OSPF (Open Shortest Path First) - протокол динамічної маршрутизації , заснований на технології відстеження стану каналу (link-state technology) для знаходження найкоротшого шляху, або BGP (Border Gateway Protocol) - протокол динамічної маршрутизації, що належить до класу протоколів маршрутизації зовнішнього шлюзу, для забезпечення оптимальних маршрутів. ✓ **Динамічна адаптація**. Здатність системи адаптуватися до змін у

		мережній топології, автоматично вибираючи оптимальні маршрути.

Стратегії оптимізації для забезпечення стабільності та швидкодії мережі спрямовані на покращення ефективності та продуктивності мережі, забезпечуючи стабільність та швидкодію в обслуговуванні користувачів та оптимізації бізнес-процесів.

7. Роль мережного моніторингу в виявленні проблем.

Системи моніторингу та їхні можливості:

Узагальнене визначення систем моніторингу – системи моніторингу представляють собою комплекс програмних та апаратних засобів, спрямованих на збір, аналіз та візуалізацію даних щодо стану мережі.

- **Можливості систем моніторингу:**
  - ✓ **Реальний час**. Здатність спостерігати за подіями в режимі реального часу, що дозволяє оперативно реагувати на неполадки.
  - ✓ **Логування та архівація**. Збереження інформації про минулі події для подальшого аналізу та виявлення закономірностей.
  - ✓ **Візуалізація**. Подання даних у зручній формі, що полегшує розуміння стану мережі.
  - ✓ **Сповіщення та тривоги**. Автоматизоване сповіщення про аномалії чи небезпеку для оперативного реагування.

Метрики у мережному моніторингу представляють числові показники, що вимірюють різні аспекти роботи мережі, такі як пропускна здатність, затримка, втрати пакетів тощо.

- **Переваги використання метрик для виявлення аномалій**:
  - ✓ **Раннє виявлення проблем**. Метрики надають можливість визначити зміни в стані мережі ще до того, як користувачі відчують наслідки.
  - ✓ **Об'єктивність**. Використання кількісних метрик дозволяє об'єктивно визначити рівень продуктивності та ідентифікувати аномалії.
  - ✓ **Тренди та аналіз**. Збір та аналіз метрик протягом тривалого періоду дозволяє виявляти тренди та передбачати можливі проблеми.
- **Приклади метрик для виявлення аномалій**:
  - ✓ **Кількість втрачених пакетів**. Зростання цієї метрики може свідчити про проблеми з надійністю мережі.
  - ✓ **Затримка (Latency)**. Великі затримки можуть вказувати на проблеми у мережних шляхах чи обладнанні.

- ✓ **Використання пропускної здатності**. Забігання до пікових значень може призвести до перевантаження та погіршення швидкодії.

Мережний моніторинг, з використанням систем та метрик, дозволяє ефективно виявляти та аналізувати аномалії у мережі, покращуючи її стабільність та швидкодію.

8. Виявлення типових проблем через мережний моніторинг

- **Відмови обладнання та перевантаження мережі** можуть призвести до втрати доступу до ресурсів, зниження продуктивності та незадовільного користування мережними послугами.

Виявлення відмов обладнання через мережний моніторинг виконуються за допомогою:

- ✓ **Метрики пропускної здатності** - загальне зменшення пропускної здатності може свідчити про перевантаження.
- ✓ **Втрати пакетів**. Збільшення втрат пакетів може бути індикатором проблеми на рівні обладнання.

Дії для виправлення:

- ✓ **Масштабування ресурсів**. Збільшення обладнання чи оптимізація його роботи для уникнення відмов**.**
- ✓ **Моніторинг завантаження**. Систематичне вивчення пропускної здатності та завантаження обладнання для передбачення можливих проблем.

- **Мережні конфлікти та невірна конфігурація** можуть призводити до втрати з'єднання, помилок у передачі даних та порушень безпеки.

Виявлення через мережний моніторинг:

- ✓ **Логи та журнали**. Зафіксовані помилки, невдалі спроби з'єднання чи зміни у конфігурації вказують на можливі конфлікти чи помилки.
- ✓ **Метрика втрат пакетів**. Збільшення втрат пакетів може бути пов'язано з конфліктами мережних адрес або проблемами конфігурації.

Дії для виправлення:

- ✓ **Аудит конфігурації**. Регулярне перевіряння конфігурації пристроїв для виявлення можливих конфліктів.
- ✓ **Використання інструментів вирішення конфліктів**, які виявляють та вирішують конфлікти мережних адрес та інші конфліктні ситуації.

У табл. 3.3 наведено основні стратегії усунення проблем та відновлення працездатності мережі, включаючи підходи до

автоматизованого виявлення й реагування на інциденти, а також планування і впровадження патчів та оновлень.

Таблиця 3.3

Основні стратегії усунення проблем та відновлення працездатності мережі

Автоматизовані засоби виявлення та реагування.	**Планування та впровадження патчів та оновлень.**
Використання систем автоматизованого моніторингу для постійного відстеження стану мережі та виявлення аномалій.	Регулярне планування та визначення необхідних оновлень та патчів для всієї мережної інфраструктури.
Встановлення автоматичних заходів реагування на виявлені аномалії без необхідності втручання адміністратора.	Систематичне виконання процесу впровадження оновлень та патчів, забезпечуючи безпеку та ефективність мережі.
✓ **Швидке реагування**. Автоматизовані системи негайно виявляють проблеми та вживають заходів для їх усунення. ✓ **Мінімізація людського фактору**. Зменшення можливості помилок, пов'язаних з людським втручанням, завдяки автоматизації.	✓ **Закриття вразливостей**. Вчасне впровадження оновлень та патчів дозволяє закривати вразливості та запобігати атакам. ✓ **Покращення функціональності**. Оновлення можуть також включати нові функції та покращення, що поліпшують роботу мережі.
Приклади автоматизованих засобів: ✓ **SNMP-моніторинг**. Використання SNMP (Simple Network Management Protocol) для автоматичного моніторингу та керування мережним обладнанням. ✓ **Системи автоматизованого виявлення інцидентів**. Використання SIEM (Security Information and Event Management) для виявлення та реагування на інциденти безпеки.	**Дії для виправлення**: ✓ **Розробка графіка оновлень**. Створення графіка для визначення регулярності та порядку впровадження оновлень. ✓ **Тестування перед впровадженням**. Перевірка оновлень на тестових системах перед їх впровадженням в продакшен-середовище.

Обидві стратегії спрямовані на забезпечення ефективного усунення проблем та відновлення роботи мережі, зменшуючи вплив негативних факторів на її стабільність та безпеку.

9. Забезпечення безпеки мережі

Забезпечення безпеки мережі виконується ефективним виявленням та реагуванням на атаки та вразливості в мережі [40]. У табл. 3.4 наведено порівняльну характеристику підходів до забезпечення мережної безпеки, зокрема методів виявлення атак і вразливостей, а також моніторингу потоків даних для визначення аномальної активності.

Таблиця 3.4

Порівняння підходів до забезпечення мережної безпеки

• **Виявлення атак та вразливостей.**	• **Моніторинг потоків даних для виявлення ненормативної активності.**
Встановлення систем виявлення вторгнень (IDS) та систем виявлення аномальної поведінки (AD) для постійного моніторингу та виявлення незвичайних подій у мережі.	Встановлення систем, які аналізують об'єм та напрямок мережного трафіку для виявлення ненормативної чи підозрілої активності.
Проведення регулярних сканувань вразливостей для виявлення слабких місць у мережній інфраструктурі та програмних засобах.	Спостереження за змінами у поведінці мережі, що може вказувати на можливі загрози чи атаки.
Переваги: ✓ **Раннє виявлення загроз**. Системи виявлення дозволяють реагувати на потенційні загрози до їх реалізації. ✓ **Моніторинг в реальному часі.** Забезпечення моніторингу та виявлення подій в реальному часі для ефективного реагування на атаки.	**Переваги:** ✓ **Деталізована аналітика.** Можливість виявлення складних та хитрих атак через деталізовану аналітику потоків даних. ✓ **Зменшення помилкових тривог.** Використання аналітичних інструментів для відсіювання помилкових тривог та концентрації на реальних загрозах.
Дії для виправлення: ✓ **Постійне оновлення баз відомостей**. Забезпечення актуальності баз відомостей	**Дії для виправлення:** ✓ **Впровадження систем аналізу поведінки.** Використання систем, які аналізують не лише

систем виявлення вторгнень та сканерів вразливостей. ✓ **Аналіз логів.** Регулярний аналіз логів для виявлення аномалій та відстеження потенційно шкідливих дій.	сигнатури атак, але й зміни у звичайному поведінці. ✓ **Інтеграція з системами виявлення вторгнень.** Спільна робота систем аналізу потоків та систем виявлення вторгнень для комплексного захисту.

У табл. 3.5 представлено основні завдання та вимоги до мережного моніторингу, які є ключовими для забезпечення стабільної, безпечної та ефективної роботи мережної інфраструктури.

Таблиця 3.5

Основні завдання та вимоги до мережного моніторингу

Відстеження трафіку та ресурсів	✓ **Моніторинг використання мережних ресурсів.** Вимірювання пропускної здатності, визначення використання широкосмугових каналів, ідентифікація основних джерел трафіку. ✓ **Аналіз трафіку для оптимізації пропускної спроможності**. Виявлення надмірної або неефективної використання мережних ресурсів.
Виявлення та вирішення проблем	✓ **Автоматизоване виявлення несправностей**. Моніторинг для виявлення неполадок в роботі мережі, таких як відмови обладнання чи проблеми зі з'єднанням. ✓ **Швидке реагування на виникнення проблем**. Автоматичні повідомлення та сповіщення про невірну роботу чи зміни у стані мережі.
Моніторинг безпеки мережі	✓ **Виявлення атак та вразливостей**. Аналіз трафіку для ідентифікації можливих атак або вразливостей в системі. ✓ **Моніторинг потоків даних для виявлення ненормативної активності**. Визначення аномальної чи підозрілої активності у мережі.
Забезпечення високої доступності	✓ **Виявлення та усунення відмов**. Визначення можливих точок відмов та розробка стратегій для уникнення або усунення їхніх наслідків. ✓ **Моніторинг навантаження**. Визначення пікового навантаження та планування резервних ресурсів для запобігання перевантаженням.
Забезпечення відповідності	✓ **Збір та аналіз даних для відповідності нормативам**. Використання мережного

до вимог і стандартів	моніторингу для перевірки відповідності безпековим стандартам та регулятивам. ✓ **Архівація та зберігання даних**. Забезпечення збереження даних моніторингу для подальшого аналізу та вирішення можливих питань безпеки чи відповідності.

Загальна мета мережного моніторингу полягає в забезпеченні стабільності, безпеки та ефективності мережної інфраструктури, а також в оперативному реагуванні на будь-які проблеми чи загрози.

10.Системи моніторингу трафіку

Існують як програмні так і апаратні рішення[51]. Деякі з них перелічено у табл. 3.6.

Таблиця 3.6

Програмні та апаратні рішення моніторингу

	Найбільш відомий приклад програмного рішення моніторингу трафіку – **Wireshark**. Відкритий програмний інструмент для аналізу мережного трафіку. Дозволяє захоплювати та аналізувати пакети даних у реальному часі. Можливості: ✓ Детальний аналіз пакетів, включаючи дані кожного рівня мережного стеку. ✓ Фільтрація та пошук пакетів за різними критеріями. ✓ Підтримка різних протоколів, включаючи TCP, UDP, IP, ICMP та інші.
	Приклад апаратного рішення моніторингу трафіку - **Cisco NetFlow**. Апаратний засіб, розробленим для пристроїв Cisco. Можливості: ✓ Захоплення та аналіз метаданих про трафік для статистики та аналізу. ✓ Визначення трафіку за допомогою різних параметрів, таких як IP-адреси, порти, протоколи тощо. ✓ Моніторинг трафіку в режимі реального часу та створення звітів.

	Аналізатор пакетів **SolarWinds Network Performance Monitor (NPM)**. Програмне рішення для моніторингу та аналізу мережного трафіку. Можливості: ✓ Аналіз пропускної здатності мережі та виявлення проблем у її роботі. ✓ Моніторинг трафіку на різних рівнях, включаючи рівні додатків та користувачів. ✓ Сповіщення про аномалії та відстеження статистики.
	Аналізатор пакетів **PRTG Network Monitor**. Програмне рішення для моніторингу та аналізу мережного трафіку. Можливості: ✓ Захоплення та аналіз пакетів даних для виявлення аномалій. ✓ Моніторинг пропускної здатності та визначення завантаження мережі. ✓ Створення звітів та графіків для аналізу та вивчення тенденцій.

Обидва типи рішень (програмні та апаратні) та аналізатори пакетів надають адміністраторам мережі можливості ефективно моніторити трафік, виявляти аномалії та розуміти, як використовується мережна інфраструктура.

Системи моніторингу мережі (NMS)

Системи моніторингу мережі (Network Monitoring System) призначені для моніторингу та аналізу мережного трафіку. Вони забезпечують видимість у реальному часі продуктивності мережі та подій безпеки. Рішення NMS дозволяють організаціям визначати потенційні загрози та вразливі місця, перш ніж впливати на їхні мережі.

Рішення для моніторингу мережі надає інформацію про стан мережі в будь-який момент часу. Це допомагає NA виявляти такі проблеми, як сповільнення або збої. Крім того, це дозволяє їм вжити заходів для вирішення проблем, перш ніж вони завдадуть шкоди.

Переваги системи моніторингу мережі (NMS):

- Сповіщення в режимі реального часу.
  Миттєве сповіщення, коли щось піде не так.
- Віддалений доступ.
  Перевірка стану мережі з будь-якої точки світу.
- Безпека.
  Відстеження, що роблять користувачі у вашій мережі.
- Продуктивність.
  Відстежувати, наскільки швидко працює мережа.
- Масштабованість.
  Додавання хостів до мережі, не вимагає зусиль про масштабованості.
- Економічна ефективність.
  Не доведеться купувати кілька одиниць обладнання, бо використовується завжди одне ядро системи.

Це лише деякі переваги системи керування мережею, а кожне конкретне програмне забезпечення керування мережею має багато інших переваг. У табл. 3.7 наведено програмні NMS

Таблиця 3.7

Програмні NMS

Назва	Тип	Функціональність
Nagios Core Nagios	Відкрита платформа	✓ Моніторинг доступності та продуктивності пристроїв та сервісів. ✓ Сповіщення про аномалії та генерація звітів. ✓ Розширена система плагінів для різноманітності моніторингу.
Zabbix ZABBIX	Відкрита платформа	✓ Моніторинг різноманітних пристроїв та служб за допомогою агентів. ✓ Збір та аналіз лог-файлів та статистики. ✓ Автоматичне виявлення пристроїв у мережі та їх параметрів.
PRTG Network Monitor PRTG NETWORK MONITOR	Комерційна платформа	✓ Спостереження за станом пристроїв, здатності мережі та роботи служб. ✓ Моніторинг ресурсів, таких як CPU, пам'ять та диски. ✓ Аналіз мережного трафіку та інформації про пропускну здатність.
SolarWinds Network Performance Monitor (NPM)	Комерційна платформа	✓ Моніторинг пропускної здатності та виявлення аномалій. ✓ Аналіз даних на різних рівнях, включаючи додаткові рівні та користувачів.

		✓ Система оповіщень та генерація звітів для аналізу та прогнозування.
Dynatrace	Комерційна платформа	✓ Моніторинг реального часу та аналіз роботи додатків та сервісів. ✓ Виявлення та діагностика аномалій у роботі програмного забезпечення. ✓ Аналіз взаємодії між компонентами додатків та їх взаємозалежності.
Zenoss	Відкрита платформа	✓ Моніторинг доступності та продуктивності мережних пристроїв. ✓ Автоматизована інвентаризація мережного обладнання. ✓ Аналіз та візуалізація даних зібраних з різних джерел.
Cacti	Відкрита платформа	✓ Графічне відображення статистики мережного трафіку. ✓ Моніторинг системних параметрів пристроїв. ✓ Створення власних шаблонів для моніторингу.
Munin	Відкрита платформа	✓ Автоматичне створення графіків для моніторингу різних параметрів. ✓ Простий інтерфейс для відслідковування змін в мережі. ✓ Повідомлення про аномалії та невідповідності.
Prometheus	Відкрита платформа	✓ Моніторинг метрик з різних компонентів мережної інфраструктури. ✓ Збір, візуалізація та аналіз даних. ✓ Підтримка реєстрів відкритих метрик (OpenMetrics).
Sematext Monitoring	Комерційна платформа	✓ Аналіз продуктивності та доступності пристроїв. ✓ Моніторинг додатків та інфраструктури в хмарних середовищах. ✓ Автоматизовані оповіщення та аналітика.
The Dude from Mikrotik	Відкрита платформа	✓ Мапування мережі та відслідковування пристроїв. ✓ Моніторинг доступності та стану мережних об'єктів. ✓ Візуалізація та аналіз трафіку.

11.Переваги використання сучасних технологій

Відношення до автоматизації мережного моніторингу за допомогою штучного інтелекту.

- **Позитивні аспекти:**
  - **Ефективність.** Штучний інтелект може виявити аномалії в мережі навіть тоді, коли вони неочікувані або важко визначити за допомогою традиційних методів.
  - **Швидкість.** Автоматизовані системи штучного інтелекту можуть виявляти та реагувати на проблеми в реальному часі, що дозволяє швидше вирішувати проблеми.
- **Виклики та обмеження:**
  - **Навчання.** Штучний інтелект вимагає навчання на великих обсягах даних, а для деяких аспектів мережного моніторингу може бути складно забезпечити достатньо великий та репрезентативний набір даних.
  - **Безпека.** Використання штучного інтелекту у мережному моніторингу може викликати питання щодо безпеки та конфіденційності даних.
- **Роль ШІ у виявленні аномалій:**
  - **Аналіз великих обсягів даних**. Штучний інтелект може ефективно аналізувати великі обсяги мережних даних та виявляти аномалії за короткий час.
  - **Моделювання поведінки.** Використання алгоритмів машинного навчання для створення моделей поведінки мережі та виявлення відхилень від звичайного.

Можливість автоматизації вирішення проблем за допомогою ШІ.

- **Автоматичні реакції.** На основі виявлених аномалій система штучного інтелекту може виробляти автоматичні реакції, такі як вимкнення атакуючого пристрою або переналаштування мережних параметрів.
- **Оптимізація.** Штучний інтелект може автоматично оптимізувати роботу мережі, реагуючи на зміни у завантаженості чи інших умовах.

Існують численні успішні практичні реалізації використання штучного інтелекту в мережному моніторингу та виявленні аномалій. Деякі приклади наведено у табл. 3.8.

Таблиця 3.8

Приклади використання штучного інтелекту в мережному моніторингу та виявленні аномалій

Приклад	**Опис**	**Результати**

Cisco Stealthwatch	Використовує аналіз мережного трафіку, машинне навчання та штучний інтелект для виявлення аномалій у мережі та потенційних загроз безпеці.	Покращена виявлення загроз та зниження часу реакції на інциденти.
Darktrace	Використовує технологію машинного навчання для неперервного моніторингу мережі та виявлення аномальної активності.	Виявлення ранніх стадій атак та аномалій, що допомагає у запобіганні серйозним кіберзагрозам.
Splunk IT Service Intelligence (ITSI)	Використовує аналітику та машинне навчання для агрегації даних з різних джерел та виявлення аномалій в мережному середовищі.	Покращена обізнаність про мережні події та їх вплив на бізнес-процеси.
IBM QRadar	Використовує аналіз поведінки, машинне навчання та алгоритми виявлення аномалій для моніторингу мережі та виявлення потенційних загроз.	Ефективне виявлення та відправлення сповіщень про можливі загрози для швидкого реагування.

Перелічені платформи та рішення служать прикладами успішних використань штучного інтелекту в мережному моніторингу, де вони допомагають виявляти аномалії, забезпечують захист від загроз та підвищують загальну безпеку мережних інфраструктур. Зауважте, що дані практики можуть змінюватися з часом, і рекомендується перевіряти найновіші відомості.

Автоматизовані засоби вирішення проблем у мережному моніторингу

Розглянемо кілька категорій автоматизованих засобів вирішення проблем у мережному моніторингу та їхні приклади.

Системи автоматичного виявлення та реагування (IDS/IPS)

IDS: система виявлення інтрузій (Intrusion Detection System) призначена для виявлення незвичайної або підозрілої активності в мережі або на окремих комп'ютерах.

IPS: система запобігання інтрузіям (Intrusion Prevention System) вживає заходів для запобігання атакам або інтрузіям, включаючи блокування атакуючого трафіку чи інші заходи безпеки.

для виявлення та запобігання кіберзагрозам, таким як атаки на мережний периметр або невірна конфігурація систем безпеки.

IDS виявляє аномалії та потенційні загрози в мережі.

IPS реагує на виявлені загрози, блокуючи атакуючий трафік або іншим чином запобігаючи невірній активності.

Розробникик - різні компанії, що надають IDS/IPS-рішення. Наприклад, Snort є відкритим рішенням, а Cisco, Palo Alto Networks, Check Point та Fortinet пропонують комерційні IDS/IPS-продукти у своїх сімействах мережних засобів безпеки.

Системи автоматизованого виявлення та відновлення (AIDR)

Automated Incident Detection and Response, як зрозуміло з назви☺, використовується для автоматизованого виявлення випадків відмов у роботі та відновлення роботи мережних компонентів.

Ця категорія систем використовується для виявлення та відновлення випадків відмов у роботі апаратного забезпечення або мережних служб.

AIDR автоматично виявляє аномалії у роботі мережних елементів та відновлює роботу шляхом переключення на резервні мережні шляхи або використання інших резервних ресурсів.

AIDR може представляти собою внутрішній розвиток конкретної організації або може бути продуктом від різних постачальників. Зазвичай, це спеціалізовані рішення в галузі кібербезпеки, але точний розробник може різнитися.

Системи автоматизованого моніторингу пропускної здатності (NPM)

Network Performance Monitoring використовується для моніторингу та аналізу пропускної здатності мережі, виявлення недоліків та оптимізації її роботи.

Категорія працює на виявлення недоліків у пропускній здатності мережі та планування її оптимізації.

Системи NPM автоматично моніторять пропускну здатність, виявляючи місця заторів або недоліки, і можуть надавати рекомендації з оптимізації.

Існує багато різних NPM-рішень, які можуть бути відкритими чи комерційними. Деякі популярні рішення включають SolarWinds Network Performance Monitor, PRTG Network Monitor, Cisco Prime Infrastructure та інші.

Системи автоматичного реагування на аномалії мережі

Концепція систем автоматичного виявлення та реагування на аномалії включає в себе використання різних технологій та підходів, таких як аналіз поведінки мережі, машинне навчання, алгоритми виявлення аномалій, та автоматизовані процеси реагування на виявлені проблеми.

Виявлення та реагування на аномалії в роботі мережі, такі як зміни у поведінці пристроїв чи несподівані великі навантаження.

Автоматизовані системи аналізують дані з пристроїв та мережі, виявляють аномалії та можуть автоматично виконувати відповідні заходи, такі як зміна конфігурації чи резервне планування мережі.

Конкретних розробників чи продуктів, які однозначно відповідають цій назві, може бути кілька. Також варто відзначити, що іноді під "системою автоматичного реагування на аномалії" може розумітися певна конфігурація різних продуктів для виявлення та вирішення аномалій, а не самостійний продукт.

Системи автоматичної зміни конфігурацій (NCCM)

Network Configuration and Change Management відповідає за зберігання, відслідковування та автоматизоване управління змінами конфігурацій мережних пристроїв для забезпечення стабільності та безпеки мережі.

Виявлення та усунення проблем, пов'язаних з невірною конфігурацією мережних пристроїв.

NCCM виявляє невірні або небезпечні конфігурації та може автоматично виконувати зміни для відновлення стабільності та безпеки мережі.

Деякі компанії, що спеціалізуються в управлінні мережною конфігурацією, включають SolarWinds NCM (Network Configuration Manager), Cisco Prime Infrastructure, Infoblox NetMRI та інші.

Хоча автоматизовані засоби вирішують багато проблем, в деяких випадках важливо залучити адміністраторів мережі (NA) або інших фахівців для вирішення складних проблем, які можуть вимагати аналізу глибокого рівня або врахування специфічних деталей конфігурацій.

Висновки до розділу

Мережний моніторинг є ключовим елементом сучасних мережних інфраструктур, надаючи організаціям засоби для ефективного управління, забезпечення безпеки та оптимізації роботи мережі.

Мережний моніторинг значно впливає на забезпечення стабільності, безпеки та ефективності мережних інфраструктур та дозволяє операторам отримувати цінні дані про стан мережі для прийняття обґрунтованих рішень та вчасного виявлення аномалій.

Завдання мережного моніторингу включають в себе виявлення аномалій, оптимізацію пропускної здатності, забезпечення безпеки та вчасну реакцію на проблеми. Вимоги полягають у високій точності, швидкодії та масштабованості систем моніторингу.

У цьому розділі описано різноманітні засоби та технології, такі як SNMP, флейви (sFlow, NetFlow), аналізатори пакетів, системи виявлення і запобігання інтрузіям (IDS/IPS) та інші, які використовуються для збору різноманітних мережних даних.

Загальною тенденцією є постійний розвиток та вдосконалення засобів моніторингу для відповіді на розширені потреби сучасних мережних інфраструктур. Мережний моніторинг є критичним елементом для забезпечення продуктивності та безпеки мережі в умовах стрімкого технологічного розвитку.

Контрольні запитання

1) Що таке мережний моніторинг і яке його основне значення у сучасних мережах?
2) Назвіть основні завдання мережного моніторингу.
3) Які технології використовуються для збору мережних даних?
4) Що таке політики Quality of Service (QoS) у керуванні трафіком?
5) У чому полягає метод traffic shaping?
6) Для чого застосовується deep packet inspection (DPI)?
7) Яку функцію виконує load balancing у мережах?
8) Що таке IDS/IPS і яку проблему вони вирішують?
9) Яке призначення систем AIDR (Automated Detection and Recovery)?
10) Що дозволяє виявити метрика втрати пакетів?
11) Які фактори впливають на виникнення джиттера (jitter)?
12) Як Content Delivery Network (CDN) підвищує швидкодію мережі?
13) Які метрики застосовуються для виявлення відмови обладнання в мережі?
14) У чому полягає взаємозв’язок між швидкістю та стабільністю мережі?
15) У яких одиницях вимірюється швидкість мережі?
16) Які фактори впливають на швидкість роботи мережі?

Розділ 4. Протоколи збору трафіку

1. Вступ до протоколів збору трафіку

Мережні протоколи відіграють ключову роль у вдосконаленні ефективності мережі, забезпечуючи збір та аналіз важливих статистичних даних. Вони дозволяють мережним адміністраторам відстежувати трафік, ідентифікувати проблеми та аномалії, оптимізувати ресурси та приймати обґрунтовані рішення для поліпшення загальної ефективності мережного середовища.

Протоколи мережного моніторингу є необхідною складовою загальної стратегії управління мережею. Їх використання дозволяє забезпечити стабільність, безпеку та ефективність мережі. Ці протоколи забезпечують засоби для отримання об'єктивної інформації про мережний трафік та його характеристики, що дозволяє розробляти та впроваджувати ефективні стратегії управління ресурсами, планування мережі та виявлення проблем. У результаті, ці протоколи сприяють підвищенню надійності та продуктивності мережі, що відповідає загальним цілям управління мережею.

Протоколи FLOW – це сімейство протоколів, які використовуються для експорту інформації про маршрутизацію та комутацію з мережних пристроїв. Ця інформація може використовуватися для моніторингу мережі, аналізу трафіку та планування мережі [6], [3].

Функціональні можливості:

- **Моніторинг мережі**. Дозволяють виконувати моніторинг мережного трафіку та виявлення проблем з мережею.
- **Аналіз трафіку**. Можуть використовуватися для аналізу мережного трафіку та визначення його характеристик.
- **Планування мережі**. Дозволяють виконувати планування мережі та прогнозування потреб у пропускній здатності.

Типи протоколів FLOW:

- **NetFlow**. Розроблений Cisco Systems і є найпоширенішим протоколом FLOW.
- **IPFIX**. Протокол є стандартом IETF і може використовуватися для експорту даних FLOW з різних типів мережних пристроїв. IETF - це скорочення від "Internet Engineering Task Force" (Інженерна робоча група Інтернету) - організація, що займається розробкою та стандартизацією протоколів Інтернету.
- **sFlow**. Розроблений компанією InMon Corporation - американською компанією, що спеціалізується на моніторингу мережного трафіку і аналітиці мережної продуктивності. Може використовуватися для експорту даних FLOW з різних типів мережних пристроїв [24].

Використання NetFlow та sFlow є невід'ємною частиною сучасних мережних технологій, оскільки ці протоколи надають важливі статистичні дані для ефективного управління мережним середовищем, але розповідь про мережні протоколи була б неповною без загального огляду протоколів збору трафіку.

2. Протокол збору трафіку SNMP

Протокол простого мережного моніторингу SNMP – протокол прикладного рівня, який є частиною протоколу TCP/IP. Він дозволяє керувати продуктивністю мережі, знаходити й усувати проблеми, планувати зростання мережі. Він збирає статистику по трафіку до кінцевого хоста через пасивні датчики, які реалізуються разом з маршрутизатором.

Головні розробники та організації, що брали участь у створенні SNMP:

📌 IETF (Internet Engineering Task Force) – міжнародна організація, що займається розробкою та стандартизацією мережних протоколів, у тому числі SNMP.

📌 Marshall T. Rose – один із ключових авторів SNMP, а також основних RFC, які визначають цей протокол.

📌 Jeffrey D. Case, Keith McCloghrie, Martin Schoffstall, James R. Davin – група експертів, що працювала над створенням SNMPv1.

📌 OSI Management Framework – на початку SNMP розглядався як тимчасове рішення, поки не буде завершена більш складна система управління мережею в моделі OSI (CMIP – Common Management Information Protocol), але врешті-решт SNMP став стандартом де-факто.

Існують дві версії (SNMP v1 і SNMP v2), SNMP v2 побудований на SNMP v1 і пропонує ряд удосконалень, таких як додавання операцій з протоколами. Стандартизований ще один, додатковий варіант версії SNMP. Версія 3 (SNMP v3) який відрізняється від перших двох підвищеною безпекою використання.

Загалом, для протоколу SNMP притаманні три ключові компоненти: керовані пристрої (Managed Devices), агенти (Agents) та системи управління мережею (NMSs – Network Management Systems) (рис. 4.1).

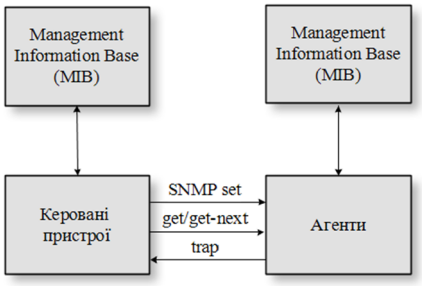


Рис. 4.1. Архітектура SNMP

Керовані пристрої включають в себе SNMP-агента і можуть складатися з маршрутизаторів, комутаторів, концентраторів, персональних комп'ютерів, принтерів і інших елементів, подібних до цих. Вони несуть відповідальність за збір інформації та роблять її доступною для системи управління мережею (NMS).

Агенти включають в себе програмне забезпечення, яке володіє інформацією з управління, і переводять цю інформацію в форму, сумісну з SNMP. Вони закриті для пристрою управління.

Системи управління мережею (NMS) виконують додатки, які займаються моніторингом і контролем пристроїв управління. Ресурси процесора і пам'яті, які необхідні для управління мережею, надаються NMS. Для будь-якої керованої мережі повинна бути створена хоча б одна система управління. SNMP може діяти виключно як NMS, або агент, або може виконувати свої обов'язки або ін.

Існує 4 основних команди, що використовуються SNMP NMS для моніторингу і контролю керованих пристроїв: **читання, запис, переривання і операції перетину.**

- ✓ **Операція читання** розглядає змінні, які зберігаються керованими пристроями.
- ✓ **Команда запису** змінює значення змінних, які зберігаються керованими пристроями.
- ✓ **Операція перетину** отримує інформацію, які змінні керованих пристроїв підтримуються, і збирає інформацію з підтримуваних таблиць змінних.
- ✓ **Операція переривання** використовується керованими пристроями для того, щоб повідомити NMS про настання певних подій.

Іншими словами, SNMP використовує 4 протокольні операції в порядку дії: **Get, GetNext, Set і Trap**.

- ✓ **Get** використовується, коли NMS видає запит на інформацію для керованих пристроїв. SNMPv1-запит складається з заголовка повідомлення і одиниці даних протоколу (PDU). PDU – повідомлення містить інформацію, яка необхідна для успішного виконання запиту, який буде або отримувати інформацію від агента, або ставити значення в агента. Керований пристрій використовує SNMP агентів, розташованих в ньому, для отримання необхідної інформації і потім посилає повідомлення NMS, з відповіддю на запит. Якщо агент не володіє будь якою інформацією стосовно запиту, він нічого не повертає.
- ✓ **GetNext** буде отримувати значення наступного примірника об'єкта.
- ✓ **Set.** Для NMS також можливо надсилати запит (операція Set), коли встановлюється значення елементів без агентів.
- ✓ **Trap.** Коли агент повинен повідомити NMS – події, він буде використовувати операцію Trap.

SNMP – протокол рівня додатків, який використовує пасивні сенсори, щоб допомогти адміністратору простежити за мережним трафіком і продуктивністю мережі, SNMP може бути корисним інструментом для мережного адміністратора, він створює можливість для загрози безпеки, тому що він позбавлений можливості автентифікації.

Моніторинг та управління по SNMP передбачає три режими роботи з пристроями:

- **Збір лише важливих значень**. В цьому випадку розглядаються тільки ті змінні, дані яких використовуються під час моніторингу. Такий підхід дозволяє заощадити обчислювальні і мережні ресурси.
- **Збір всіх значень з MIB-директорії.** Система збирає значення всіх змінних, для яких існують MIB-опису. Режим забезпечує збір повної інформації про роботу пристрою. Проаналізувавши ці дані, зможемо вирішити, які аспекти потребують моніторингу.
- **Збір всіх значень, доступних при скануванні.** Система збирає всі доступні SNMP-змінні, незалежно від того, є їх опису чи ні. Нерозпізнані величини представляються в табличній формі і просто не мають зрозумілих описів і імен. Такий підхід дозволяє отримати всі дані, що надаються пристроєм, а потім підібрати для невизначених значень MIB з метаінформацією.

Протягом багатьох років регулярний моніторинг мережі виконувався протоколом SNMP, який надає огляд IT-інфраструктури, надаючи мережним адміністраторам інформацію про наявність його компонентів і, як зрозуміло з наявності різних версій, SNMP пройшов декілька еволюційних етапів, кожен з яких приносив покращення в безпеці, ефективності та функціональності.

Розглянемо детальніше версії SNMP

- **SNMPv1 (1988).** Це перша версія протоколу, яка отримала широке поширення завдяки простоті впровадження.

Основні особливості:

Використовує "community strings" для аутентифікації (public – для читання, private – для запису).

Підтримує базові операції:

Get – отримання значення певного параметра.

Set – зміна значення параметра.

GetNext – отримання наступного значення в таблиці OID (об'єктний ідентифікатор).

Trap – надсилання повідомлення про події агентом до сервера SNMP Manager.

Недоліки SNMPv1:

Відсутність механізму шифрування → всі дані передаються у відкритому вигляді. Community strings (рядки аутентифікації) передаються незахищеними.

Немає підтримки групових запитів → обмежена ефективність при зборі великих обсягів даних.

Через ці обмеження SNMPv1 більше не рекомендується для використання в сучасних мережах.

- **SNMPv2 (1993–1996) –** Покращена продуктивність Щоб усунути обмеження SNMPv1, було розроблено SNMPv2, який приніс значні покращення у швидкості збору даних і функціональності.

Основні покращення у SNMPv2:

Операція GetBulk – дозволяє отримувати великі обсяги даних за один запит (замість послідовного GetNext в SNMPv1).

Розширені SNMP Traps – агент може надсилати більше деталей про подію.

Покращена обробка помилок – введено більш детальні коди помилок для кращої діагностики.

Різновиди SNMPv2:

- ✓ **SNMPv2p (Party-based SNMPv2)** – перша версія з покращеною безпекою, але була складною у впровадженні, тому не стала популярною.
- ✓ **SNMPv2u (User-based SNMPv2)** – ще одна спроба покращити безпеку, яка теж не отримала широкого використання.
- ✓ **SNMPv2c (1996)** – спрощена версія, яка повернулася до community strings як в SNMPv1, але з усіма покращеннями продуктивності SNMPv2.

Недоліки SNMPv2c:

Все ще відсутнє шифрування, аутентифікація здійснюється через відкриті рядки (community strings).

Community strings можна перехопити, що робить SNMPv2c небезпечним для використання в незахищених мережах.

- **SNMPv3 (2002)** – Безпечний SNMP. Щоб виправити проблеми безпеки попередніх версій, було створено SNMPv3, який вперше додав шифрування, аутентифікацію та контроль доступу.

Основні покращення SNMPv3:

- ✓ Аутентифікація користувачів – введена система ідентифікації через облікові записи.
- ✓ Шифрування трафіку – використання AES або DES для захисту SNMP-запитів і відповідей.
- ✓ Контроль доступу – можливість обмежити права користувачів (read-only, read-write, no-access).

- ✓ Захист від атак "людина посередині" (MITM) – захист від підміни SNMP-повідомлень.

Три рівні безпеки в SNMPv3:

- ✓ **noAuthNoPriv** – без аутентифікації та шифрування (аналог SNMPv2c).
- ✓ **authNoPriv** – з аутентифікацією (MD5 або SHA), але без шифрування.
- ✓ **authPriv** – з аутентифікацією та шифруванням (AES або DES).

Переваги SNMPv3:

- ✓ Надійний рівень безпеки (шифрування + автентифікація).
- ✓ Гнучка система керування доступом.
- ✓ Підходить для корпоративних і критичних інфраструктур.

Недоліки SNMPv3:

Ускладнена конфігурація порівняно з SNMPv1/v2c та значно вищі вимоги до ресурсів через шифрування.

Хронологія розвитку SNMP:

1988 – SNMPv1 (RFC 1157)

розроблений як тимчасове рішення для моніторингу мереж TCP/IP.

1993-1996 – SNMPv2 (RFC 1441-1452)

покращує продуктивність, але версія SNMPv2p не стає популярною.

1996 – SNMPv2c (RFC 1901-1908)

стає основною версією SNMPv2, але без покращеної безпеки.

1996 – SNMPv2p (RFC 1441-1452)

Party-based SNMPv2, нова модель безпеки на основі "party-based" доступу. Залишився складним у реалізації, тому не став популярним.

1996 – SNMPv2u (RFC 1909-1910)

User-based SNMPv2, альтернативна версія, що спрощує механізм безпеки, орієнтуючись на користувачів. Не отримала широкого розповсюдження через відсутність зворотної сумісності зі SNMPv1.

2002 – SNMPv3 (RFC 3411-3418)

додає автентифікацію, шифрування та контроль доступу.

SNMPv2p та SNMPv2u залишилися проміжними версіями, не отримали широкого розповсюдження, а SNMPv2c став основною версією SNMPv2.

Для повноцінного вивчення основ моніторингу за допомогою SNMP існує велика кількість програмних інструментів, які можуть бути корисними на різних етапах освоєння цієї технології. Вибір конкретного інструменту залежить від рівня складності досліджуваних мережних сценаріїв, а також від специфіки роботи з різними версіями SNMP.

1. Інструменти для базового моніторингу SNMP, іншими словами, інструменти що підходять для побудови простих навчальних стендів і початкового знайомства з SNMP доцільно використовувати такі рішення:

- **snmpd (Net-SNMP)** – один із найпоширеніших SNMP-агентів, який дозволяє організувати базове опитування мережних вузлів та отримувати інформацію про їхній стан. Підтримує SNMPv1, SNMPv2c та SNMPv3.Основні файли:
  - ✓ snmpd.conf – файл конфігурації агента;
  - ✓ snmpd – основний виконуваний файл (демон);
  - ✓ snmpwalk, snmpget, snmpset – утиліти для взаємодії з агентом.
- **MiniSNMPd** – легковаговий SNMP-демон, який займає мінімальний обсяг пам'яті та є зручним для запуску у віртуалізованих середовищах або на пристроях з обмеженими ресурсами. Основні файли:
  - ✓ minisnmpd.conf – конфігураційний файл;
  - ✓ minisnmpd – виконуваний файл демона.

2. Інструменти для моделювання складних SNMP-мереж. У випадках, коли потрібно працювати зі складними мережними середовищами, необхідно використовувати спеціалізовані симулятори SNMP, які дають змогу створювати віртуальні пристрої для тестування моніторингових систем. До таких відносяться:

- **SolarWinds SNMP Simulator** – потужний комерційний інструмент для емуляції SNMP-пристроїв. Дозволяє налаштовувати відповіді на SNMP-запити, що спрощує тестування SNMP-моніторингу без фізичних пристроїв. Основні файли:
  - ✓ snmp-simulator.exe – виконуваний файл для запуску симулятора;
  - ✓ snmp-devices.cfg – файл конфігурації емуйованих пристроїв.
- **OpenNMS SNMP Mock** – безкоштовний SNMP-емулятор, що дозволяє створювати віртуальні SNMP-пристрої для тестування систем моніторингу. Основні файли:

snmp-mock.jar – виконуваний файл у форматі Java Archive;
mock-config.xml – файл налаштувань параметрів емуляції.

3. Інструменти для роботи з SNMPv3. Якщо необхідно працювати з безпечними версіями SNMP, зокрема з SNMPv3, який підтримує автентифікацію та шифрування, варто використовувати спеціалізовані інструменти:

- **iReasoning SNMP Manager** – графічний клієнт для роботи з усіма версіями SNMP, включаючи SNMPv3. Забезпечує зручний інтерфейс для виконання SNMP-запитів та аналізу відповідей. Основні файли:

- ✓ iReasoning_SNMP_Manager.exe – виконуваний файл програми;
- ✓ snmpv3-config.xml – файл конфігурації для SNMPv3.

➢ **Net-SNMP** – універсальний набір утиліт для роботи зі SNMP, що підтримує SNMPv3 з автентифікацією та шифруванням. Основні файли аналогічні snmpd.

4. Інструменти для тестування SNMP-трапів. SNMP-трапи – це механізм повідомлення про події в мережі, коли SNMP-агент надсилає повідомлення SNMP-менеджеру про зміну стану або виникнення помилки. Для тестування SNMP-трапів використовуються такі утиліти:

➢ **SNMPTT (SNMP Trap Translator)** – спеціалізована програма для отримання, фільтрації та обробки SNMP-трапів, що перетворює їх на більш зрозумілий формат. Основні файли:
- ✓ snmptt.ini – головний конфігураційний файл;
- ✓ snmptt – виконуваний файл.

Вибір інструмента для моніторингу SNMP залежить від конкретних завдань:
Для початкового вивчення підходять snmpd (Net-SNMP) або MiniSNMPd.
Для моделювання складних мереж можна використовувати SolarWinds SNMP Simulator або OpenNMS SNMP Mock.
Якщо потрібна підтримка SNMPv3, доцільно застосовувати iReasoning SNMP Manager або Net-SNMP.
Для роботи з SNMP-трапами рекомендовано SNMPTT.

Для Windows немає snmpd, оскільки відсутній нативний Net-SNMP-агент, але є альтернативні варіанти:

➢ **Windows SNMP Service** (вбудований у Windows) виконує роль SNMP-агента (snmpd на Linux). SNMP Service вмикається через "Увімкнення або вимкнення компонентів Windows". Налаштовується через services.msc.

➢ **Net-SNMP для Windows - SNMP-агент (snmpd)** та клієнтські утиліти (snmpwalk, snmpget), що можуть бути завантажується з офіційного сайту Net-SNMP: https://net-snmp.sourceforge.io та встановлені на сервері або робочій станції Windows.

3. Протокол збору трафіку IPFIX

IPFIX (IP Flow Information Export) – це стандартний протокол збору, експорту та аналізу мережного трафіку, розроблений IETF (Internet Engineering Task Force). Він базується на NetFlow v9, але з розширеними можливостями, що забезпечують універсальний формат передачі даних про мережні потоки. IPFIX дозволяє маршрутизаторам, комутаторам, мережним

зондами та іншим пристроям передавати детальну статистику про трафік у вигляді потоків на сервери збору даних для подальшого аналізу [7].

Хронологія розвитку IPFIX

📌 2001 рік – Cisco розробила NetFlow v9 – протокол для експорту мережного трафіку, який став основою для IPFIX.

📌 2004 рік – робоча група IETF почала розробку універсального стандарту, заснованого на NetFlow v9.

📌 2008 рік – офіційна специфікація IPFIX була затверджена у RFC 5101 та RFC 5102.

📌 2013 рік – вийшло доповнення RFC 7011, яке удосконалило гнучкість та ефективність IPFIX.

📌 2020-ті роки – IPFIX активно використовується в SIEM-системах, мережному моніторингу та аналізі загроз.

Стандартна реалізація протоколу складається з трьох типових функціональних компонент, що використовуються для аналізу IPFIX:

- **IPFIX Exporter** - маршрутизатор, комутатор, що зондує, або хост-програмний агент з підтримкою IPFIX, який відстежує статистику ключів і іншу інформацію про потік IP – пакетів і генерує записи потоку, які вміщені в UDP і відправляються в збирач потоків.
- **Collector IPFIX** - додаток, відповідальний за прийом пакетів записів потоку, опрацювання даних із записів потоку, попередню обробку і збереження запису потоку від одного або декількох експортерів потоку.
- **IPFIX Analyzer** - програмний додаток, який надає табличні, графічні та інші інструменти і візуалізації, щоб мережні оператори та інженери могли аналізувати дані потоку для різних випадків використання, включаючи моніторинг продуктивності мережі, усунення неполадок і планування пропускної здатності.

На рис. 4.02 наведена «страшна» схема розширеної архітектури IPFIX. Ми не будемо детально на ній зупинятися, бо вона базується на NetFlow v9, а опис цих протоколів нас чекає попереду.

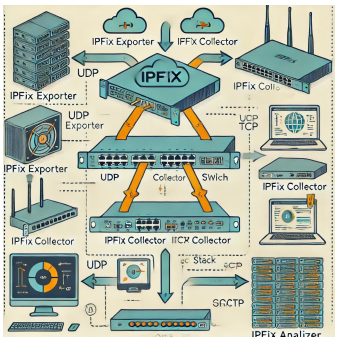


Рис. 4.02. Розширена архітектура IPFIX

Основні інструменти для роботи з IPFIX

✓ Колектори (IPFIX Collectors):

✓ nfdump/nfsen – потужний інструмент для збору та аналізу NetFlow/IPFIX.

- ✓ IPFIXcol – один із популярних колекторів з підтримкою SQL-баз даних.
- ✓ Elasticsearch + Logstash + Kibana (ELK) – застосовується для аналізу трафіку у великих інфраструктурах.

✓ Аналізатори (IPFIX Analyzers):

- ✓ ntopng – інтерактивний веб-аналізатор мережного трафіку.
- ✓ Wireshark – підтримує IPFIX для розширеного аналізу мережних потоків.

Plixer Scrutinizer – комерційне рішення для детального аналізу трафіку.

Переваги та недоліки IPFIX

Переваги:

- ✓ Гнучкість – підтримує різні типи даних та адаптується до потреб адміністраторів.
- ✓ Стандартність – є відкритим та офіційно затвердженим IETF.
- ✓ Розширені можливості порівняно з NetFlow (наприклад, підтримка нестандартних полів).
- ✓ Висока деталізація трафіку – дозволяє відстежувати не лише IP-адреси, а й метрики часу, QoS, VPN-трафік тощо.

Недоліки:

- ✓ Вимагає більше ресурсів для обробки даних, ніж NetFlow v9, на якому базується.
- ✓ Ускладнена конфігурація – потребує ретельного налаштування експортерів та колекторів.
- ✓ Не всі комерційні пристрої повністю підтримують IPFIX (деякі виробники залишаються на NetFlow).

IPFIX є потужним інструментом для детального аналізу мережного трафіку та моніторингу загроз, що успадкував багато переваг NetFlow, але водночас став більш універсальним та гнучким стандартом. Завдяки підтримці відкритих форматів та розширених можливостей, IPFIX активно використовується в мережній безпеці, кібермоніторингу та аналізі продуктивності мереж, проте має і ряд суттєвих недоліків.

4. Протокол збору трафіку JFlow

JFlow —мережний протокол збору та аналізу потоків даних, розроблений компанією Juniper Networks. Він використовується для моніторингу IP-трафіку в маршрутизаторах, комутаторах і брандмауерах, дозволяючи адміністраторам отримувати детальну

статистику про використання смуги пропускання, аналізувати продуктивність мережі та виявляти потенційні загрози безпеці.

Історія розвитку та концепція

Протокол ***JFlow*** (рис. 4.03) є фактично еквівалентом ***Cisco NetFlow*** і працює за аналогічним принципом. Концепція збору потоків даних була розроблена у 1990-х роках для моніторингу продуктивності мереж і аналізу використання ресурсів. Основна ідея полягає у відстеженні послідовності IP-пакетів, які мають однакові параметри (IP-адреси джерела/одержувача, порти, протокол та QoS), що дозволяє ефективно агрегувати інформацію та зменшувати навантаження на мережу.

Juniper Networks впровадила JFlow як власний варіант NetFlow, який інтегрується в їхні пристрої та операційну систему Junos OS. Протокол працює на принципі експорту потокових даних з пристроїв до сервера збору, де виконується їх аналіз і візуалізація.

Принцип роботи JFlow

JFlow (рис.4.03) базується на вибірці (sampling) потоків даних і працює за таким алгоритмом:

- ✓ Маршрутизатор/комутатор збирає інформацію про IP-пакети, що проходять через нього.
- ✓ На основі певних параметрів (IP-адреси, порти, протокол, QoS) визначаються потоки.

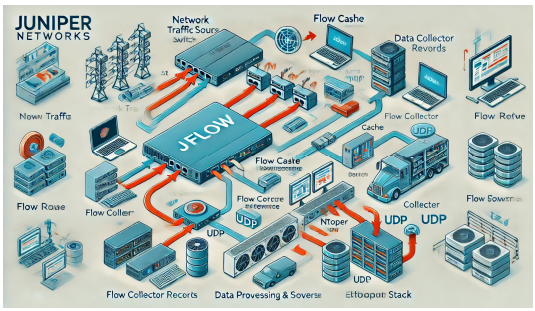


Рис. 4.03. Розширена архітектура JFlow

- ✓ Потік вважається активним, доки триває передача даних. Якщо потік завершується або досягає таймауту, інформація про нього експортується до сервера збору даних.
- ✓ Сервер аналізує отримані дані, агрегує їх і формує звіти про використання мережі.

Протокол експортує зібрані дані через UDP до серверів моніторингу, таких як:

- ✓ PRTG Network Monitor
- ✓ Elasticsearch (через Logstash)
- ✓ ntopng
- ✓ Flow-tools
- ✓ SolarWinds NetFlow Traffic Analyzer

Відмінності JFlow від інших flow-протоколів

Протокол	Виробник	Відмінності
NetFlow	Cisco	Оригінальний протокол, має версії 5, 7, 9, IPFIX (стандарт IETF)
JFlow	Juniper	Аналог NetFlow, сумісний із версією 5/9, використовується у Junos OS
sFlow	HP, Extreme Networks	Використовує вибіркову вибірку (sampling) замість збору всіх потоків
IPFIX	Стандарт IETF	Розширений NetFlow v9 з можливістю додаткових полів
NetStream	Huawei	Власний аналог NetFlow від Huawei
RFlow	Mikrotik	Спрощена версія NetFlow для RouterOS

Основна відмінність JFlow від sFlow полягає в методі збору даних:

- ✓ **JFlow (NetFlow)** записує весь потік та зберігає інформацію у кеші перед передачею.
- ✓ **sFlow** працює на базі вибірки (sampling), що зменшує навантаження, але може призводити до втрати точності [25].

Переваги JFlow

- ✓ Висока деталізація — збір повної інформації про сесії дозволяє отримувати точні аналітичні дані.
- ✓ Сумісність — підтримує NetFlow v5/v9, тому легко інтегрується з існуючими рішеннями для аналізу трафіку.
- ✓ Оптимізація продуктивності — працює на рівні ядра Junos OS, що забезпечує швидкість обробки.
- ✓ Мінімальне навантаження на мережу — UDP-експорт потоків не впливає на продуктивність трафіку.

Недоліки JFlow

- ✓ Прив’язка до Juniper — технологія використовується лише на пристроях Juniper Networks.
- ✓ Необхідність зовнішнього сервера — для зберігання та обробки даних потрібне додаткове програмне забезпечення.
- ✓ Не всі версії Junos підтримують NetFlow v9 — у старих пристроях може бути доступна лише v5.

JFlow є потужним інструментом для збору мережного трафіку, що надає детальну статистику про використання пропускної здатності. Він особливо корисний для аналізу продуктивності, виявлення аномалій у

мережі, оптимізації QoS та забезпечення безпеки. Завдяки сумісності з NetFlow, його можна легко інтегрувати з популярними системами моніторингу. Однак, через обмеження на використання тільки в екосистемі Juniper, організації з різнорідним мережним обладнанням можуть віддати перевагу універсальним рішенням, таким як IPFIX або sFlow [16].

5. Визначення та основні характеристики протоколу NetFlow.

Історія та походження протоколу NetFlow

Протокол NetFlow є важливою складовою інфраструктури мережного моніторингу та обліку трафіку. Розроблений компанією Cisco Systems, він визначається як мережний протокол, спрямований на облік мережного трафіку, і став фактичною стандартною практикою в цьому напрямку.

- **Походження та початок розвитку.** Історія NetFlow налічує кілька десятиліть, з початку робіт над ним у середині 1990-х років. Cisco, лідер в галузі мережних технологій, почала впроваджувати цей протокол як рішення для відслідковування трафіку в мережах. В основі створення NetFlow лежала необхідність забезпечити адміністраторам мережі засобами для збору та аналізу інформації про мережний трафік, що стало все більш важливим у зростаючому світі комп'ютерних мереж.
- **Стандартизація та поширення.** З часом NetFlow став не лише виробничим рішенням Cisco, але й стандартом для багатьох інших виробників мережного обладнання, таких як Juniper і Enterasys. Це внесло великий вклад у його поширення та застосування в інших мережних середовищах, роблячи його ефективним інструментом для різних типів компаній та організацій.
- **Функції та можливості.** Аналізуючи дані, які надає NetFlow, адміністратори мереж можуть отримувати детальну інформацію про джерело та призначення трафіку, клас обслуговування та причини перевантаження. Це відкриває можливості для вчасного виявлення проблем, вдосконалення ефективності мережі та забезпечення безпеки.
- **Версії та розвиток.** Існують кілька версій протоколу NetFlow, але на 2011 рік найбільш поширеними є версії 5 і 9. Версія 5 визначає формат обміну даними та стала дуже поширеною у мережних пристроях Cisco. Версія 9, яка є більш розширеною і забезпечує більше можливостей, також послужила основою для створення відкритого стандарту під назвою IPFIX (Internet Protocol Flow Information eXport - експорт інформації про потоки IP). IPFIX визначається як стандарт, що регулює механізми експорту інформації про потоки IP, що дозволяє використовувати аналогічні можливості не лише на обладнанні Cisco, але й на різних платформах та пристроях, роблячи протокол більш універсальним та гнучким.

NetFlow, розпочавши свій шлях у світі мережних технологій, став важливим стандартом для обліку та аналізу мережного трафіку, що визначає його великий вплив на розвиток інфраструктури моніторингу та управління мережею.

Принцип дії протоколу NetFlow.При відкритті чергового сеансу передачі даних на мережному обладнанні формується інформація про даний сеанс, так званий потік (flow). Відомості про потік включають кількість переданих байтів, вхідний і вихідний інтерфейси для сеансу, IP-адреси відправника / одержувача, порти відправника / одержувача, номер протоколу IP, параметри QoS і т.д. Потоки акумулюються на мережному пристрої і відправляються колектору NetFlow в датаграму UDP. Колектор NetFlow агрегує отриману інформацію, проводить аналіз і формує зручні для сприйняття звіти і графіки. Один з популярних колекторів NetFlow – NetFlow Analyzer, але існують колектори.. Протокол NetFlow дозволяє отримати повну картину трафіку в каналах. Можна переглянути якісний склад трафіку (IP-адреси, порти, додатки) в будь – якому сегменті мережі, а також оцінити, яку частку пропускної здатності каналу (у відсотковому відношенні) займає той чи інший потік.

Основні характеристики протоколу NetFlow. Протокол NetFlow, як ключовий інструмент для обліку та аналізу мережного трафіку, відрізняється рядом важливих характеристик, що визначають його ефективність та застосування в різних мережних середовищах.

- ✓ **Передача даних.** Передача даних в рамках протоколу NetFlow відбувається за допомогою створення та експорту "потоків" з різних мережних пристроїв. Потік - це набір даних, що визначає комунікаційну сесію між двома пристроями в мережі. Кожен потік містить ключову інформацію, таку як джерело і призначення трафіку, порти, протокол та інші атрибути. Ці дані надсилаються до NetFlow-коллектора в реальному часі, що дозволяє адміністраторам отримувати негайну звітність про мережний стан та трафік.

  Однією з ключових переваг передачі даних за допомогою потоків є зменшення обсягу інформації, що передається, забезпечуючи ефективну використання пропускної здатності мережі. Крім того, це робить NetFlow ефективним інструментом для моніторингу великих та високовитратних мереж.
- ✓ **Підтримувані пристрої.** Початково розроблений компанією Cisco, NetFlow вперше з'явився на мережних пристроях цього виробника. Проте з розвитком та зростанням популярності, інші виробники мережного обладнання, такі як Juniper, Enterasys, та інші, також почали імплементувати підтримку NetFlow в свої пристрої.

Сучасна екосистема NetFlow охоплює широкий спектр обладнання, включаючи рутери, комутатори, файрволи та інші мережні пристрої. Це забезпечує адміністраторам мереж можливість використання NetFlow в різних топологіях мережі та конфігураціях, що робить його важливим інструментом у багатьох сценаріях.

- ✓ **Архітектура.** Для збору інформації про трафік за протоколом NetFlow потрібні такі компоненти (рис.4.04):
  - ➢ **Сенсор.** Збирає статистику з трафіку, що проходить через нього. Зазвичай це L3-комутатор або маршрутизатор, хоча можна використовувати і сенсори, що окремо стоять, одержують дані шляхом дзеркалювання порту комутатора.

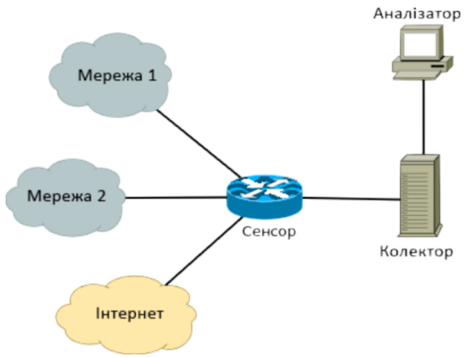


Рис. 4.4. Архітектура NetFlow

  - ➢ **Колектор.** Збирає отримані від сенсора дані та поміщає їх у сховище.
  - ➢ **Аналізатор.** Аналізує зібрані колектором дані та формує придатні для читання людиною звіти (часто у вигляді графіків).

- ✓ **Формат даних** NetFlow визначається версією протоколу. Дві найбільш використовувані версії - 5 і 9 - відрізняються у своїх підходах до структури даних.
  - ➢ **Версія 1.** Протоколу NetFlow на сьогоднішній день вже застаріла і не використовується.
  - ➢ **Версії 2-4.** Ніколи не включалися в Cisco IOS, і, відповідно, не підтримуються.
  - ➢ **Версія 5**. Використовує фіксований формат запису, що містить основну інформацію про потік, таку як IP-адреси джерела та призначення, порти, протокол та об'єм переданого трафіку. Версія 5 була доповнена вимогами про номери автономних систем, використовуваних в протоколі граничного шлюзу (BGP), і номер потоку. Ця версія досі використовується, якщо немає потреби у додатковій інформації, яка надається більш пізніми версіями протоколу та дозволяє ефективно передавати базові дані для аналізу та моніторингу.
  - ➢ **Версія 6.** Ніколи не включалися в Cisco IOS. Не підтримується.
  - ➢ **Версія 7.** Стала розвитком версії 5 і вона стала підтримуватися в серії комутаторів Cisco Catalyst.
  - ➢ **Версія 8.** У цій версії була введена можливість агрегації даних, що актуально при великих обсягах даних протоколу NetFlow.

- **Версія 9.** Забезпечує більшу гнучкість, дозволяючи включати додаткові атрибути в запис про потікза рахунок істотного збільшення кількісті полів в порівнянні з версією 5. Ці додаткові поля дозволяють розширити і уточнити інформацію, що проходить в мережному потоці. Наприклад, версія 9 включає інформацію 2-го рівня мережної моделі: підтримка IPv6, групових (Multicast) розсилок, параметри протоколів MPLS, BGP і інше. Але основне нововведення версії 9 – це використання шаблонів, що забезпечує легке розширення протоколу за рахунок використання нових типів шаблонів даних.
- **IPFIX.** Важливою особливістю є те, що на основі версії 9 був розроблений стандарт IPFIX (Internet Protocol Flow Information eXport), який став відкритим стандартом для обміну інформацією про потоки IP. IPFIX створений як стандарт, що може бути використаний на різних платформах та обладнанні, незалежно від виробника. Це дозволяє різним вендорам реалізовувати цей стандарт у своїх пристроях та програмних рішеннях, забезпечуючи уніфікований підхід до обміну інформацією про потоки IP.

✓ **Опис протоколу.** NetFlow використовує UDP ("User Datagram Protocol" - протокол користувацького дейтаграмного протоколу) або SCTP ("Stream Control Transmission Protocol" - протокол керування потоком передачі) для передачі даних про трафік колектору. Як правило, колектор слухає порт **2055, 9555** чи **9995**. Сенсор виділяє з проходить трафіку потоки , що характеризуються наступними параметрами:

- Адреса джерела;
- Адреса призначення;
- Порт джерела для UDP та TCP;
- Порт призначення для UDP та TCP;
- Тип та код повідомлення для ICMP ;
- Номер протоколу IP ;
- Мережний інтерфейс (параметр ifindex SNMP);
- IP Type of Service .

Потоком вважається набір пакетів, які у одному напрямі. Коли сенсор визначає, що потік закінчився (за зміною параметрів пакетів, або скидання TCP-сесії), він відправляє інформацію в колектор. Залежно від налаштувань він також може періодично відправляти в колектор інформацію про потоки, що все ще йдуть.

Зібрана інформація надсилається у вигляді записів, що містять такі параметри (для версії 5):

- номер версії протоколу;
- Номер запису;
- Вхідний та вихідний мережний інтерфейс;

- Час початку та кінця потоку;
- Кількість байт та пакетів у потоці;
- Адреса джерела та призначення;
- Порт джерела та призначення;
- номер протоколу IP;
- значення Type of Service;
- Для TCP-з'єднань - всі прапорці, що спостерігаються протягом з'єднання;
- Адреса шлюзу;
- Маски підмережі джерела та призначення.

Версія 9 також підтримує додаткові поля, такі як заголовки IPv6 , мітки потоків MPLS та адресу шлюзу BGP . Деякі сенсори можуть також підтримувати номер автономної системи .

Якщо використовується UDP, то втрачений через проблеми з мережею запис не буде отриманий колектором. Колектор може визначити втрати пакетів за значеннями номера запису, які за стандартом мають бути зростаючими.

Якщо сенсором виступає мережний пристрій (маршрутизатор чи комутатор), то економії ресурсів NetFlow включають лише тих інтерфейсів, у яких хочуть збирати статистику.

Для економії ресурсів процесора також застосовується "sampled NetFlow". У цьому випадку сенсор аналізує не всі, а кожен n-ий пакет, де n може бути заданим адміністративно або випадковим чином, що вибирається. При використанні sampled NetFlow одержувані значення не точними, а оціночними [58].

Характеристики **NetFlow** визначають його як потужний інструмент для моніторингу та управління мережею. Здатність передачі даних в реальному часі, підтримка на різних мережних пристроях та гнучкість формату даних роблять його ефективним засобом для отримання детальної інформації про мережний трафік та забезпечення адекватного управління мережею.

6. Походження та розвиток sFlow

sFlow – це відкритий стандарт для моніторингу трафіку в мережах, призначений для надання детальної статистики та аналізу в реальному часі (рис.4.05). Розглянемо походження та розвиток цього протоколу, який визначає його сутність та значення в сучасних мережних технологіях.

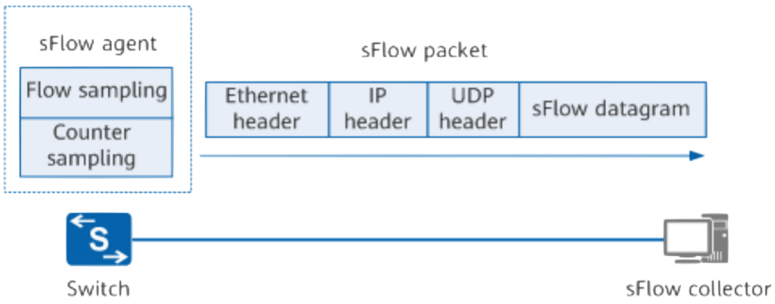


Рис. 4.05. Архітектура sFlow

- **Історія та Походження sFlow.** sFlow був вперше представлений компанією InMon Corporation на початку 2000-х років. Головною метою створення було розробити універсальний стандарт, який дозволяв багатьом виробникам мережного обладнання реалізувати моніторинг трафіку на своїх пристроях.
  Важливою перевагою sFlow є його відкритість та здатність працювати на різних мережних пристроях, незалежно від їхнього виробника чи моделі. Це відмінно від інших протоколів, які можуть бути властивими лише певному обладнанню.
- **Універсальність та розвиток sFlow.** sFlow визначається своєю універсальністю та розширюваністю. Завдяки принципу вибіркового вибору та зразків, sFlow може збирати статистику лише з частини трафіку, зменшуючи навантаження на мережу та забезпечуючи широкий обхват аналізу.
  Розвиток sFlow включав у себе не лише поширення його підтримки серед різних виробників, а й постійне вдосконалення стандарту. Додаткові функції та можливості були додані в нових версіях протоколу, що розширює його функціональність та адаптує для змін у мережних технологіях.
- **Активне та пасивне відстеження.** Однією з ключових особливостей sFlow є можливість як активного, так і пасивного відстеження. У режимі активного відстеження sFlow взаємодіє з мережним обладнанням, надсилаючи зразки трафіку на аналіз. У пасивному режимі sFlow збирає дані з вже існуючого трафіку без втручання у роботу мережі.
- **Відкриті реалізації та стандартизація.** Однією з ключових переваг sFlow є наявність відкритих реалізацій для різних операційних систем та платформ. Це дозволяє використовувати sFlow на серверах, комутаторах, маршрутизаторах та іншому обладнанні.
  Стандарт сфокусований на відкритості і стандартизації, і це робить його придатним для широкого спектру застосувань. Багато вендорів використовують sFlow як основний інструмент для мережного моніторингу, забезпечуючи адміністраторам мереж високий рівень контролю та аналізу мережного трафіку.

Унікальні характеристики sFlow.

sFlow, як протокол для моніторингу трафіку в мережах, вирізняється своїми унікальними характеристиками, які роблять його ефективним та гнучким інструментом для адміністраторів мереж [17].

- **Активне та пасивне відстеження.**
  - ✓ **Активне відстеження** визначається можливістю sFlow взаємодіяти з мережним обладнанням для збору та надсилання статистики трафіку. Цей режим дозволяє вибірково збирати дані з різних точок мережі, адаптуючися до конкретних вимог адміністратора. Наприклад, можливість визначити певні порти чи пристрої для моніторингу, дозволяє вимірювати трафік на обраній частині мережі, що є важливим для точного аналізу та управління.
  - ✓ **Пасивне відстеження** означає можливість sFlow збирати дані з існуючого трафіку без втручання у роботу мережі. Цей режим особливо корисний для аналізу та моніторингу трафіку без його переривання чи впливу на нього. Наприклад, адміністратор може використовувати пасивне відстеження для виявлення аномалій або моніторингу пропускної здатності без значного впливу на нормальну роботу мережі.

У табл. 4.1 розглянуто конкретні сценарії застосування активного та пасивного відстеження sFlow [9]:

Таблиця 4.1

Сценарії застосування активного та пасивного відстеження sFlow

Активне відстеження	Пасивне відстеження
Припустимо, що адміністратор мережі хоче детально моніторити трафік на конкретному комутаторі в центральному офісі. Використовуючи активне відстеження sFlow, він може налаштувати цей комутатор для вибіркового збору даних і періодично надсилання їх до централізованого моніторингового сервера. Такий підхід дозволяє зосередитися на конкретних аспектах мережі та забезпечити більш ефективний моніторинг.	Припустимо, що мережний адміністратор хоче вивчати поведінку користувачів на сервері без втручання у конфігурацію самого сервера. Використовуючи пасивне відстеження sFlow на суміжньому комутаторі, він може збирати статистику трафіку, визначаючи найактивніші порти та типи послуг, які використовуються. Це дозволяє адміністраторові отримати інформацію про використання ресурсів сервера без прямого взаємодії з самим сервером.

Розділ присвячений також NetFlow, у якого немає явної підтримки активного відстеження. Однак деякі виробники мережного обладнання можуть використовувати власні реалізації, які можуть включати елементи активного моніторингу, але це вже залежить від конкретного обладнання та виробника.

sFlow, навпаки, більше орієнтований на активний моніторинг. sFlow дозволяє відстежувати трафік на активних пристроях, які підтримують цей протокол. sFlow відправляє відібрані зразки пакетів з активних мережних пристроїв до аналізатора, де їх можна використовувати для моніторингу та аналізу. Отже, в цьому випадку sFlow має більшу спрямованість на активний моніторинг, тоді як NetFlow, який зазвичай працює у пасивному режимі, може включати елементи активного моніторингу в окремих виробників обладнання.

Формат даних.

Формат даних sFlow визначає, як саме інформація про трафік буде структурована та передаватися. sFlow використовує JSON-подібний формат для опису даних, що робить його легким для розуміння та обробки. Кожний зразок (sFlow sample) містить інформацію про конкретний аспект мережного трафіку, такий як IP-адреса, порти, протоколи, типи послуг та інші важливі атрибути.

Приклад структури даних sFlow в форматі JSON-подібного запису:

```
{
  "agent_address": "192.168.1.1",
  "input_port": 1,
  "output_port": 2,
  "packet_count": 1000,
  "byte_count": 120000,
  "protocol": "TCP",
  "source_ip": "10.0.0.1",
  "destination_ip": "10.0.0.2",
  "source_port": 5000,
  "destination_port": 80
}
```

У цьому прикладі зразок sFlow містить інформацію про передачу трафіку з IP-адреси 10.0.0.1 на IP-адресу 10.0.0.2 через TCP-з'єднання на порті 80. Дані про кількість пакетів та обсяг байтів також включені для повного опису трафіку.

Описаний формат даних дозволяє адміністраторам ефективно аналізувати та обробляти інформацію, зібрану за допомогою sFlow, спрощуючи взаємодію з моніторинговими системами та іншими інструментами аналізу мережного трафіку.

sFlow виникає як універсальний стандарт для моніторингу та аналізу трафіку в мережах. З його походженням від компанії InMon і активним розвитком у вигляді відкритих реалізацій та стандартів, sFlow став важливим інструментом для адміністраторів мереж, які прагнуть забезпечити ефективний моніторинг та управління своєю інфраструктурою.

7. Використання протоколів NetFlow та sFlow для збору мережних статистичних даних.

Збір даних NetFlow

- ✓ **Як працює збір даних NetFlow.** Протокол NetFlow є ефективним інструментом для збору та аналізу статистики трафіку в мережах. Розглянемо, як саме відбувається збір даних NetFlow та які процеси відбуваються під час цього процесу.

1. **Експорт даних.** Процес збору даних NetFlow розпочинається на мережних пристроях, таких як маршрутизатори чи комутатори. Кожен такий пристрій, який підтримує NetFlow, володіє таблицями потоків, в яких зберігається інформація про протоколи, IP-адреси, порти та інші атрибути трафіку.

   Коли пристрій розпізнає новий потік (у вигляді пакету або групи пакетів, що належать одному потоку), він приймає рішення про експорт цих даних до зовнішнього агенту, який може бути NetFlow-колектором або іншим пристроєм, призначеним для збору і обробки даних.

2. **Формування пакетів NetFlow.** Щоб передати інформацію про потоки до колектору, дані повинні бути упаковані в спеціальні пакети NetFlow. Ці пакети містять ключову інформацію про кожен потік, яка дозволяє аналізувати трафік на більш високому рівні.
   Наприклад, пакет NetFlow може містити наступну інформацію для кожного потоку:
   - Номер версії протоколу.
   - Номер запису.
   - Вхідний та вихідний мережний інтерфейс.
   - Час початку та кінця потоку.
   - Кількість байт та пакетів у потоці.
   - Адреса джерела та призначення.
   - Порт джерела та призначення.
   - номер протоколу IP.
   - значення Type of Service.

- Для TCP-з'єднань - всі прапорці, що спостерігаються протягом з'єднання.
- Адреса шлюзу.
- Маски підмережі джерела та призначення.

3. **Відправлення до NetFlow-колектора.** Сформовані пакети NetFlow відправляються до NetFlow-колектора за допомогою мережі. NetFlow-колектор може бути спеціальним пристроєм або програмним забезпеченням, призначеним для збору та обробки даних NetFlow.

   Цей колектор приймає пакети, дешифрує їх та зберігає інформацію у вигляді записів або бази даних для подальшого аналізу. Одним з основних завдань колектора є збереження історії потоків для подальшого вивчення та виявлення аномалій чи проблем у мережі.

4. **Аналіз та використання даних NetFlow.** Отримані дані NetFlow надають адміністраторам мережі детальний огляд того, як використовуються ресурси мережі. Інструменти аналізу можуть включати графіки трафіку, звіти про користувачів, виявлення аномалій та оптимізацію роботи мережі.
   Приклад:
   Розглянемо сценарій, де NetFlow використовується для аналізу трафіку в офісній мережі. Мережний адміністратор виявляє, що певний комп'ютер витрачає надмірно багато ресурсів мережі. Збираючи дані NetFlow, він може ідентифікувати IP-адресу цього комп'ютера, перевірити, які додатки використовуються, і при потребі вжити заходів для обмеження або оптимізації трафіку, що генерується цим пристроєм.

Висновки до розділу

Резюмуючи основні аспекти NetFlow, слід вказати на його ключові характеристики, такі як здатність збору детальних даних про мережний трафік, виявлення проблем та використання в різноманітних сценаріях. Щодо sFlow, важливо визначити його унікальні можливості, такі як активне та пасивне відстеження, які роблять його ефективним інструментом для аналізу мережі.

Використання NetFlow та sFlow у щоденній роботі NA відіграє важливу роль у підтримці ефективності та безпеки мережі. NA можуть ефективно моніторити та управляти мережним трафіком, виявляти аномалії та вирішувати проблеми в реальному часі. Збір та аналіз даних, наданих цими протоколами, робить можливим покращення продуктивності та оптимізацію інфраструктури.

Розглядаючи шляхи подальшого розвитку NetFlow та sFlow, можна звернутися до новітніх технологій, таких як підтримка швидших швидкостей передачі даних, розширення функціональності для аналізу мережних пакетів, а також адаптацію до змін у вимогах до безпеки.

Загальною тенденцією може бути інтеграція протоколів NetFlow та sFlow з іншими інструментами моніторингу та аналізу мережі, що розширює їхню функціональність та забезпечує більшу зручність в управлінні мережею.

Збір та аналіз мережних даних за допомогою NetFlow та sFlow є ключовими аспектами для успішного функціонування та ефективного управління сучасними мережами.

Контрольні запитання

1) Які основні протоколи використовуються для збору мережного трафіку?
2) Хто є розробником стандарту NetFlow?
3) Хто розробив стандарт sFlow?
4) На основі якого стандарту було розроблено IPFIX?
5) Яке призначення мають Flow-колектори?
6) Які протоколи використовує NetFlow для передачі даних до колектора?
7) Які основні компоненти входять до архітектури SNMP?
8) Назвіть основні протокольні операції SNMP.
9) Що таке пасивне відстеження трафіку?
10) Які поля можуть входити до пакету NetFlow?
11) У чому відмінність алгоритму роботи NetFlow від sFlow?
12) Які унікальні риси протоколу sFlow?
13) Для чого використовується NetFlow-колектор?
14) Які функції може виконувати NetFlow-колектор?
15) Чи є SNMP інструментом для виявлення шкідливих активностей у мережі?
16) Чи використовується SNMP як засіб flow-моніторингу?

Розділ 5. Інструменти та технології системного моніторингу

1. Огляд популярних інструментів системного моніторингу. Nagios.

Nagios

❖ **Nagios vs Icinga. Історична довідка**

У 1996 році американець Ітан Ґалстад (Ethan Galstad) розробив програму для MS-DOS, яка дозволяла перевіряти доступність серверів Novell NetWare за допомогою сторонніх програм. На основі цієї простої конструкції через три роки у 1999 році була розроблена система з відкритим вихідним кодом NetSaint, який став попередником того, що сьогодні відоме як програмне забезпечення для моніторингу Nagios. Netsaint був розроблений Ітаном для власних потреб. Таким чином, перша версія Nagios датується минулим століттям: 1999 роком.

Через 10 років (Травень 2009) на світ з'явилася Icinga (дериватив Nagios). Про причини, які змусили команду розробників Icinga на чолі з Міхаелем Люббеном (Michael Lubben) - одним з засновників Icinga та провідним розробником доповнень до Nagios, відокремитися і розпочати власний проект є цікава стаття Nagios Vs. Icinga: the real story of one of the most heated forks in free software Треба згадати, що команда Icinga, як і раніше, пересилає всі свої напрацювання до Nagios та зберігається дуже висока сумісність цих двох моніторингових систем. Насправді, Nagios має майже нескінченну кількість відгалужень: Icinga, OpsView, Op5, Centreon, Naemon, Shinken.

❖ **Що таке Nagios** – це інструмент безперервного моніторингу з відкритим вихідним кодом, який контролює мережу, програми та сервери. Він може знаходити та усувати проблеми, виявлені в інфраструктурі, та зупиняти майбутні проблеми, перш ніж вони торкнуться кінцевих користувачів. Це дає повний статус вашої IT-інфраструктури та її продуктивність [34].

❖ **Чому Nagios.** Nagios пропонує такі функції, які роблять його доступним для великої групи користувачів –

- Може контролювати сервери баз даних, такі як SQL Server, Oracle, MySQL, Postgres
- Надає інформацію про рівень програми (Apache, Postfix, LDAP, Citrix тощо. буд.).
- Забезпечує активний розвиток.
- Має відмінну підтримку від величезної активної спільноти.
- Nagios працює над моніторингом будь-якої операційної системи.
- Підтримує більшість стандартизованих мережних протоколів.

❖ **Переваги Nagios.** Nagios пропонує такі переваги для користувачів –

- Допомагає позбутися періодичного тестування.
- Виявляє збої за частки секунди, коли подія перебуває у «переривчастій» стадії.
- Знижує вартість обслуговування без шкоди продуктивності.
- Забезпечує своєчасне оповіщення персоналу та керівництва про контроль та поломки.

❖ **Архітектура Nagios.** Наступні пункти стосуються архітектури Nagios –

- Nagios має архітектуру сервер-агент.
- Сервер Nagios встановлений на хості
- Плагіни (агенти) встановлені на віддалених хостах/серверах, які мають контролюватись. Підтримуються методи безагентного моніторингу.
- Nagios посилає сигнал через процес планування для запуску плагінів на локальних / віддалених хостах / серверах.
- Плагіни збирають дані (завантаження процесора, використання пам'яті тощо. буд.) і відправляють їх у планувальник.
- Потім графіки процесу надсилають повідомлення адміністраторам та оновлюють графічний інтерфейс Nagios.

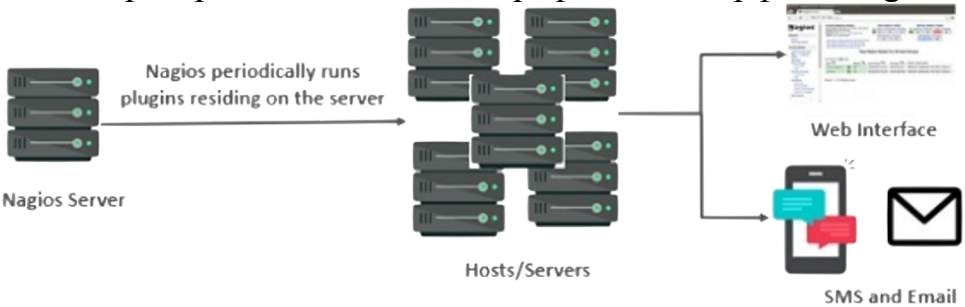


Рис. 5.01. Загальна архітектура Nagios

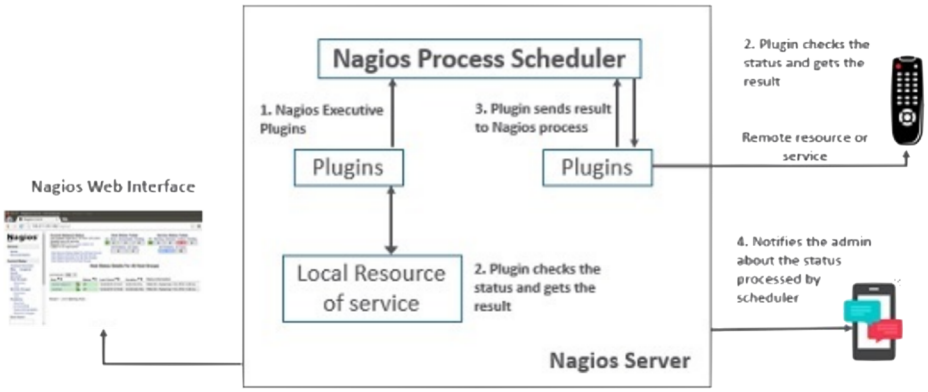


Рис. 5.02. Архітектура серверу Nagios

На рис. 5.01 та 5.02 показано загальну архітектуру та архітектуру серверу Nagios.

- Методи **безагентного моніторингу**, доступні в Nagios:
  - **SNMP (Simple Network Management Protocol)**: використовується для моніторингу мережних пристроїв, таких як маршрутизатори, комутатори та сервери, різноманітної периферії та інших пристроїв, що підтримують SNMP.
  - **WMI (Windows Management Instrumentation)**: використовується для моніторингу систем Windows.
  - **JMX (Java Management Extensions)**: використовується для моніторингу Java-додатків.
  - **HTTP/HTTPS**: використовується для моніторингу веб-серверів та інших веб-додатків.
  - **SSH**: використовується для моніторингу систем Linux та Unix.

- **Nagios – продукти.** Nagios містить лінійку різноманітних продуктів, які дозволяють налаштувати більшість схем та видів системного та мережного моніторингу. Розглянемо актуальну лінійку продуктів.
  - **Nagios XI**

Забезпечує моніторинг всіх компонентів IT-інфраструктури, таких як додатки, послуги, мережа, операційні системи і т.д. Nagios XI дає повне уявлення про вашу інфраструктуру та бізнес-процеси. Графічний інтерфейс користувача легко налаштовується, забезпечуючи гнучкість використання. Базова стандартна версія (базова це для 50 нодів) Nagios XI коштує 1295 доларів, а ціна базової (теж 50 нодів) Enterprise Edition ліцензії – 2995 доларів.

Enterprise Edition включає додаткові функції, такі як Capacity Planning Reports, SLA Reports, Bulk Modification та інші, що полегшують управління великомасштабними конфігураціями та

звітністю. Для великих організацій, які потребують моніторингу численних серверів, доступні Sitewide ліцензії, що пропонують значні знижки. Деталі можна отримати, зв'язавшись з відділом продажів Nagios. Ціни можуть змінюватися, тому рекомендується перевірити актуальну інформацію на офіційному сайті Nagios або зв'язатися відділом продажів для отримання найсвіжішої інформації.

- **Nagios Core**

Nagios Core - ядро моніторингу IT-інфраструктури. Продукт Nagios XI також ґрунтується на Nagios Core. Щоразу, коли виникає будь-яка проблема в інфраструктурі, він відправляє попередження/повідомлення адміністратору, який може швидко вжити заходів для вирішення проблеми. Цей інструмент є абсолютно безкоштовним.

- **Nagios Log Server**

Nagios Log Server робить пошук даних журналу дуже простим та легким. Усі дані журналу зберігаються в одному місці з налаштуванням високої доступності. Він може легко надсилати оповіщення, якщо будь-яка проблема виявлена в даних журналу. Він може масштабуватись до тисяч серверів, забезпечуючи велику потужність, швидкість, об'єм сховища та надійність для платформи аналізу журналів. Nagios Log Server пропонує різні ліцензійні плани, залежно від кількості інстанцій (екземплярів) та потреб вашої організації. Ціни на офіційному сайті Nagios вказані в доларах США. Ось основні ліцензійні рівні та їх вартість:

- Mid-Scale (Середній масштаб):
  - ✓ 2-інстанційна ліцензія: $5,995
  - ✓ 3-інстанційна ліцензія: $6,995
  - ✓ 4-інстанційна ліцензія: $7,995

Ці ліцензії забезпечують централізоване управління логами, підвищену швидкість запитів, резервування даних та автоматичне переключення у випадку відмови.

- Large-Scale (Великий масштаб):
  - ✓ 10-інстанційна ліцензія: ціна доступна за запитом

Цей план підходить для моніторингу значних обсягів логів з вбудованою відмовостійкістю та резервуванням.

Зверніть увагу, що Nagios не рекомендує використовувати одноінстанційні розгортання Nagios Log Server через відсутність резервування та автоматичного переключення у випадку відмови. Для отримання детальної інформації та розрахунку вартості відповідно до ваших потреб, ви можете скористатися калькулятором планів на офіційному сайті Nagios. Також, в Україні доступні локальні реселери, такі як Softkey.UA, де ви можете придбати ліцензії. Зверніть

увагу, що ціни можуть змінюватися, тому рекомендується перевіряти актуальну інформацію на офіційних ресурсах або звертатися до офіційних представників Nagios..

- **Nagios Fusion**

Nagios Fusion забезпечує централізоване представлення всієї системи моніторингу. За допомогою Nagios Fusion ви скануєте налаштування окремих серверів моніторингу для різних регіонів. Його можна легко інтегрувати з Nagios XI та Nagios Core, щоб забезпечити повну видимість інфраструктури. Ліцензія на Nagios Fusion надається на кожну інстанцію програми. Це означає, що для кожного розгортання Nagios Fusion потрібна окрема ліцензія. Водночас, немає обмежень на кількість інстанцій Nagios XI, Nagios Log Server або Nagios Core, які можна інтегрувати з кожною інстанцією Nagios Fusion. Актуальні ціни на ліцензії Nagios Fusion можуть змінюватися, тому рекомендується перевіряти їх безпосередньо на офіційному сайті Nagios або у авторизованих реселерів. Наприклад, інтернет-магазин Softkey.UA пропонує ліцензії Nagios Fusion, але для отримання точної вартості та умов придбання слід звертатися безпосередньо до представників магазину. Придбання ліцензії Nagios Fusion надає доступ до технічної підтримки, оновлень версій протягом одного року, а також до спеціального форуму підтримки для клієнтів. Це забезпечує своєчасне отримання оновлень та допомогу у вирішенні можливих питань під час використання продукту.

- **Nagios Network Analyzer**

Nagios Network Analyzer надає повну інформацію про мережну інфраструктуру адміністратору з потенційними загрозами в мережі, щоб адміністратор міг вдатися до швидких дій. Він ділиться дуже докладними даними мережі після глибокого аналізу мережі. Ліцензування цього продукту базується на кількості інстанцій (екземплярів) програмного забезпечення, які планується розгорнути. Кожна інстанція потребує окремої ліцензії, при цьому немає обмежень на кількість джерел мережного трафіку, які можна інтегрувати з однією ліцензією. Рекомендується отримувати ліцензування Nagios Network Analyzer у локальні реселерів в Україні:

- ✓ Softkey.UA: Пропонує ліцензію Nagios Network Analyzer за ціною 86 782,50 грн.
- ✓ Allsoft Ukraine: Надає ліцензію за 77 805 грн з ПДВ.

Придбання ліцензії Nagios Network Analyzer включає доступ до технічної підтримки та оновлень протягом одного року. Це забезпечує отримання останніх версій програмного забезпечення, нових функцій

та виправлень, а також доступ до спеціалізованого форуму підтримки для клієнтів..

➢ **Nagios плагіни та додатки.**

З усієї лінійки продуктів лише Nagios Core є безкоштовною версією системи моніторингу. Треба відмітити, що його потужності можна значно розширити за допомогою різноманітних плагінів та додатків, які розробляються спільнотою та іншими сторонніми розробниками. Ці додатки дозволяють користувачам налаштовувати моніторинг для різних типів ресурсів та додатків, роблячи Nagios Core більш адаптивним до конкретних потреб користувачів. Ключові можливості розширення Nagios Core безкоштовно включають:

- **Плагіни для моніторингу різних ресурсів**. Спільнота та розробники активно створюють плагіни, які додають підтримку для моніторингу різних системних ресурсів, таких як CPU, пам'ять, диск, мережа, тощо.
- **Розширення для моніторингу додатків та сервісів**. Додатки можуть додавати підтримку моніторингу конкретних додатків, служб чи пристроїв, що робить систему більш гнучкою.
- **Інтеграція з іншими інструментами**. Існує можливість інтеграції Nagios Core з іншими інструментами та сервісами, що розширює його можливості управління та відображення даних.
- **Офіційні плагіни Nagios.** Є 50 офіційних плагінів Nagios. Офіційні плагіни Nagios розробляються та підтримуються офіційною командою плагінів Nagios.
- **Плагіни спільноти**. Існує понад 3000 сторонніх плагінів Nagios, розроблених сотнями членів спільноти Nagios.
- **Плагіни користувача**. Ви також можете написати свої власні плагіни. Існують рекомендації та документація, які необхідно дотримуватися при написанні плагінів користувача.

Взагалі кажучи, спільнота надає широкий спектр безкоштовних розширень, які дозволяють адаптувати Nagios Core під конкретні вимоги користувача, роблячи його ефективним та потужним інструментом системного моніторингу.

❖ **Встановлення Nagios.** Інсталяція Nagios виконується у ОС Linux з використанням LAMP-стеку (Linux, Apache, MySQL, PHP) або LEMP-стеку (Linux, Nginx, MariaDB, PHP). Детально інсталяція актуальної версії Nagios Core описана у методичних вказівках до лабораторних робіт і може бути виконана за кілька кроків.

❖ **Файли конфігурації Nagios** знаходяться в /usr/local/nagios/etc. Давайте розберемось з функціональністю та значимістю кожного файлу.

- **nagios.cfg** Основний конфігураційний файл Nagios Core. Цей файл містить розташування файлу журналу Nagios, інтервал оновлення стану хостів та сервісів, файл блокування та файл status.dat. У цьому файлі визначено користувачів та групи Nagios, у яких працюють екземпляри. У ньому вказуються шляхи до всіх файлів конфігурації окремих об'єктів, як-от команди, контакти, шаблони, хости, тощо.
- **cgi.cfg** За замовчанням файл конфігурації CGI Nagios називається cgi.cfg. Він повідомляє CGI (Computer Generated Imagery. У буквальному перекладі воно означає «картинки, створені на комп' ютері».), де знайти основний конфігураційний файл. CGI зчитуватиме основний та основний конфігураційні файли для будь-яких інших даних, які можуть їм знадобитися. Він містить всю інформацію про користувачів та групи, а також їхні права та дозволи. Він також має шлях до всіх зовнішніх інтерфейсів Nagios.
- **resource.cfg** У цьому файлі визначаються макроси $USERx$, які, у свою чергу, можуть використовуватися для визначення команд у файлах конфігурації хоста. Макроси $USERx$ корисні для зберігання конфіденційної інформації, такий як імена користувачів, паролі тощо. Вони також зручні для зберігання шляху до плагінів та обробників подій – якщо ви вирішите перемістити плагіни або обробники подій до іншого каталогу в майбутньому, ви можете просто оновити один або два макроси $USERx$ замість зміни великої кількості визначення команд. Файли ресурсів також можуть бути використані для зберігання директив конфігурації для зовнішніх джерел даних, таких як MySQL.
- **objects/commands.cfg** Містить деякі приклади визначення команд, які можна використовувати у визначеннях хоста, служби та контакту. Ці команди використовуються для перевірки та моніторингу хостів та сервісів. Ці команди можливо запускати локально на Linux консолі.
- **objects/contacts.cfg.** Містить інформацію про контакти та групи Nagios. За замовчуванням в одному контакті вже є адміністратор Nagios.
- **objects/templates.cfg.** Файл містить деякі приклади шаблонів визначень об'єктів, на які посилаються інші визначення хоста, служби, контакту і т. д. в інших файлах конфігурації.
- **objects/timeperiods.cfg.** Містить деякі приклади визначень часових періодів, які можна використовувати у визначеннях хоста, служб, контактів та залежностей.

❖ **Особливості Nagios Core.** Nagios – інструмент моніторингу з безліччю функцій.

- Nagios Core – програмне забезпечення з відкритим кодом, безкоштовний для використання.
- Потужний механізм моніторингу, який може масштабувати та керувати тисячами хостів та серверів.
- Комплексна веб-панель моніторингу, що дозволяє побачити всі компоненти мережі та дані моніторингу.
- Він має багатотентенантні можливості, коли кілька користувачів мають доступ до панелі моніторингу Nagios.
- Він має архітектуру, що розширюється, яка може легко інтегруватися зі сторонніми додатками з декількома API.
- Nagios має дуже активну та велику спільноту з більш ніж 1 мільйоном користувачів по всьому світу.
- Система швидкого оповіщення відправляє оповіщення адміністраторам відразу після виявлення будь-якої проблеми.
- Доступні плагіни для підтримки Nagios, плагіни з кодом користувача також можуть використовуватися з Nagios.
- Він має хороший журнал та систему бази даних, що зберігає все, що відбувається в мережі з легкістю.
- Функція запобіжного планування допомагає дізнатися, коли настав час оновити інфраструктуру.

❖ **Програми Nagios.** Nagios може застосовуватись для широкого спектру застосувань.

- Моніторинг ресурсів вузла, як-от дисковий простір, системні журнали тощо.
- Моніторинг мережних ресурсів – http, ftp, smtp, ssh тощо.
- Моніторинг файлів журналів безперервно, щоб ідентифікувати інфра-проблему.
- Моніторинг Windows / Linux / Unix / веб-додатків та його стан.
- Плагін Nagios (NRPE) може здійснювати віддалений моніторинг сервісів.
- Паралельний запуск сервісних перевірок.
- Можна використовувати тунелі SSH або SSL для віддаленого моніторингу.
- Надсилати оповіщення / повідомлення електронною поштою, смс, месенджери про події.
- Формувати рекомендації щодо оновлення IT-інфраструктури.

❖ **Nagios – Хости та сервіси.** Для розуміння принципів роботи Nagios, необхідно згадати про такі поняття як хости та сервіси. Хости та сервісні конфігурації є будівельними блоками Nagios Core.

- **Хост** – це як комп'ютер; це може бути фізичний, або віртуальний пристрій.
- **Сервіс** – це служба або параметр, що використовує Nagios для перевірки певної інформації про хост.

Файл хоста чи сервісу може бути створений в каталозі або підкаталозі сервера Nagios. Відповідно у ньому визначається хост чи сервіс. Каталог, де система має «шукати» файли конфігурацій хостів та сервісів описується у nagios.cfg

Конфігураційні файли Nagos, як відмічає більшість користувачів системи, зроблені не найкращим чином. Імена конгфігураційних файлів можуть змінюватися від версії до версії. Наприклад, в одній з попередніх версій опис всіх серверів та їх сервісів рекомендувалося зберігати в одному файлі. Звичайно, що це не найкращий варіант, тому на кожен сервер – свій файл (простіше шукати і правити, якщо щось не так). На щастя, Nagios дозволяє змінювати структуру конфігів, з чим ми вже працювали у лабораторних роботах.

У методичних рекомендаціях до лабораторних робіт описано підхід до формування конфігураційних файлів хостів у Nagios, за якого для кожного хосту створюється окремий файл у відповідному каталозі, що визначається для групи хостів (наприклад Windows-сервери та Linux-сервери). Така структура забезпечує кілька важливих переваг:

- ✓ **Логічна впорядкованість**: Розміщення конфігураційних файлів у каталогах, згрупованих за певною ознакою (наприклад, за типом серверів, географічним розташуванням або роллю в інфраструктурі), дозволяє швидко знаходити необхідні файли. Це особливо важливо при масштабуванні системи моніторингу.
- ✓ **Прозорість змін**: Завдяки чіткій структурі каталогів адміністратор завжди знає, де саме потрібно внести зміни при коригуванні налаштувань моніторингу конкретного хосту. Це знижує ризик помилок і спрощує підтримку конфігурації.
- ✓ **Модульність і гнучкість**: Окремі файли конфігурації хостів легко переносити між середовищами, створювати резервні копії або відновлювати їх без необхідності редагувати великий монолітний конфігураційний файл.
- ✓ **Полегшення розподіленої роботи**: У команді адміністраторів такий підхід сприяє поділу обов'язків, де кожен член команди може відповідати за свою групу хостів, не впливаючи на решту інфраструктури.
- ✓ **Зручність автоматизації**: Структурована конфігурація полегшує створення сценаріїв автоматичного додавання, зміни або видалення хостів з моніторингу.

2. Огляд інструменту системного моніторингу Zabbix.

ZABBIX

- **Що таке Zabbix** – це відкрите програмне забезпечення для моніторингу та управління мережею, серверами, додатками та різноманітними іншими IT-ресурсами. Як інструмент системного моніторингу з відкритим вихідним кодом, Zabbix дозволяє виявляти, вирішувати та передбачати проблеми в інфраструктурі комп'ютерних систем. За допомогою різноманітних сенсорів та збірників даних, Zabbix надає повний обзор стану IT-середовища, забезпечуючи користувачам засоби для ефективного управління та підтримки продуктивності. Інсталяція Zabbix виконується у ОС Linux з використанням LAMP-стеку (Linux, Apache, MySQL, PHP), LEMP-стеку (Linux, Nginx, MariaDB, PHP) або LAPP-стеку (Linux, Apache, PostgreSQL, PHP) [{49}].
- **Історична довідка.**
  Zabbix був розроблений програмістом Олексієм Владишевим у 1998 році як проект внутрішнього програмного забезпечення. Через 3 роки, в 2001 був випущений публічно під ліцензією General Public License. Більше трьох років минуло до виходу першої стабільної версії, 1.0, яка вийшла 2004. Версія 2.0 побачила світ у травні 2012. На сьогодні офіційну підтримку Zabbix виконує ZABBIX Company з реєстрацією у м. Рига, Латвія на чолі з «батьком» цієї системи моніторингу - Alexei Vladishev. Актуальна на момент написання цього посібника версія – 7.4.

- **Zabbix дозволяє збирати різні метрики**, і це далеко не повний перелік:

  - Мережні пристрої
  - Хмарні сервіси, контейнери, віртуальні машини
  - Моніторинг операційних систем
  - Лог-файли
  - Бази даних
  - Додатки
  - Сервіси
  - IoT датчик
  - Моніторинг веб-сторінок
  - Моніторинг кінцевих точок HTTP/HTTPS
  - Підтримуючи всіх стандартних для галузі протоколів
  - Збір даних із зовнішніх кінцевих точок API

- **Два типи вбудованих високопродуктивних агентів для всіх операційних систем та апаратних платформ**.
  У системі моніторингу Zabbix для збору даних із серверів і пристроїв існує два основних типи агентів:

✓ Zabbix Agent (першої версії)

✓ Zabbix Agent 2

Давайте розглянемо їх по черзі!

1. Zabbix Agent (Classic або просто Zabbix Agent) - класичний агент, що використовується в Zabbix вже багато років. Написаний на C, відзначається стабільністю та невеликим споживанням ресурсів. Підтримує основні метрики системи: навантаження на CPU, RAM, дискову підсистему, мережну активність тощо. Працює за двома режимами:
   - ✓ Active checks — агент ініціює з'єднання із сервером та надсилає дані.
   - ✓ Passive checks — сервер ініціює з'єднання та запитує дані з агента.

   Classic Zabbix Agent обмежений у розширюваності — для складніших перевірок (наприклад, моніторинг баз даних), коли потрібні зовнішні скрипти. Виправдане використання, якщо потрібен базовий моніторинг ОС, ресурсів і служб чи, коли стабільність і низьке споживання ресурсів є критичними.
2. Zabbix Agent 2 - новіший агент, представлений у Zabbix 5.0 (LTS). Написаний на C і Go, що дозволяє значно розширити його функціонал.

   Включає плагіни для специфічних перевірок «з коробки» (наприклад, моніторинг баз даних MySQL, PostgreSQL, Docker, RabbitMQ, Apache, Nginx тощо). Підтримує асинхронну обробку запитів, що зменшує навантаження на сервер і мережу. Має зворотну сумісність із конфігурацією класичного агента. Споживає трохи більше ресурсів, ніж класичний агент (через Go-частину коду).

   Використання виправдане для розширеного моніторингу сучасних застосунків, контейнерів і баз даних, коли необхідна асинхронність і висока швидкість обробки метрик, або якщо ви хочете зменшити кількість зовнішніх скриптів за рахунок вбудованих плагінів.

Ключові відмінності між Zabbix Agent та Zabbix Agent 2

Особливість	Zabbix Agent	Zabbix Agent 2
Мова	C	C + Go
Плагінна архітектура	Немає	Є (Docker, MySQL, Apache тощо)
Тип обробки запитів	Синхронна	Асинхронна
Вплив на ресурси	Низький	Трохи вищий
Розширюваність	Обмежена	Висока

Якщо потрібно базове відстеження стану серверів, використовується класичний Zabbix Agent. Якщо налаштовується комплексний моніторинг із сучасними сервісами — Zabbix Agent 2 може суттєво полегшити ваше життя, зменшивши кількість кастомних скриптів і конфігурацій.

Агенти Zabbix мають наступні риси:

- Незначний вплив на продуктивність
- Доставка даних за принципом push або pull
- Гнучкі інтервали планування
- Легко розширюється за допомогою плагінів або зовнішніх скриптів
- Несприйнятливість до проблем зв'язку (буферизація даних у пам'яті чи диску)
- Зворотна сумісність для зручності оновлення
- Доступ до Windows WMI та лічильників продуктивності
- Можна створювати звіти на декількох серверах Zabbix для резервування та високої продуктивності
- Надійний протокол TLS або загальний ключ шифрування для обміну даними
- Розгортання агентів Zabbix у вигляді пакету, використання MSI або програми встановлення командного рядка у Windows або розгортання попередньо скомпільованих двійкових файлів агенту.

- **Використання безагентного моніторингу із застосуванням будь-яких протоколів.** Zabbix підтримує цілу низку різних протоколів для віддаленого моніторингу сервісів:
  - Веб-моніторинг
  - Синтетичний моніторинг із підтримкою сценаріїв
  - Опитування та вилов SNMP (v1/2c/3)
  - Моніторинг програм Java
  - IPMI
  - Перевірки SSH/Telnet
  - Моніторинг ODBC
  - Перевірки ICMP та TCP
  - Легко розширюється за допомогою зовнішніх скриптів або плагінів
  - Збір даних із кінцевих точок HTTP
  - Підтримка протоколів Modbus та MQTT

- **Визначення гнучких порогів.** За допомогою Zabbix можна автоматично виявляти стан проблеми у вхідному потоці даних:
  - Високопродуктивне виявлення проблем у режимі реального часу
  - Гнучкі можливості визначення
  - Розділення стану вирішення проблем і самих проблем
  - Декілька рівнів значущості

- Аналіз вихідних даних
- Захист від схлопування
- Виявлення аномалій
- Прогнозування проблем
- Виявлені проблеми можна класифікувати за допомогою тегів для розумних оповіщень
- Експорт виявлених проблем подій у режимі реального часу до сторонніх систем (Elastic, Splunk і т.д.)

Zabbix надає гнучкі розумні можливості визначення порогових значень. Хоча поріг спрацьовування може бути простим ("більше, ніж x"), користувач може використовувати всі можливості підтримуваних функцій та операторів для статистичного аналізу історичних даних .

- **Інсталяція Zabbix** може бути виконана декількома способами, зазвичай залежно від ваших потреб та технічного стеку [{57}]. Ось кілька типових методів:
  - **Використання офіційних пакунків** - надає офіційні репозиторії для багатьох популярних дистрибутивів Linux, таких як Ubuntu, Debian, CentOS, Red Hat, та інших. Ви можете встановити Zabbix, використовуючи пакунки з цих репозиторіїв за допомогою менеджера пакунків вашої операційної системи.
  - **Використання Docker** для запуску Zabbix. Завантажте образи Zabbix з Docker Hub і запустіть контейнери за допомогою команд Docker. Це дозволяє швидко розгорнути тестове середовище Zabbix або навіть вирішити виробничні завдання.
  - **Ручне встановлення**, якщо ви бажаєте більшої контролю над процесом інсталяції або плануєте використовувати специфічні конфігурації. Це зазвичай включає встановлення всіх необхідних компонентів, таких як веб-сервер, база даних та інші залежності, а потім налаштування Zabbix через веб-інтерфейс або конфігураційні файли.
  - **Використання Zabbix Appliance** - готових віртуальних машин або аплаансів, які можуть бути завантажені та запущені на віртуальних платформах, таких як VMware або Hyper-V. Це швидкий спосіб отримати робочий екземпляр Zabbix без необхідності вручну налаштовувати середовище.
  - **Використання Zabbix Cloud -** хмарні рішення, такі як Zabbix Cloud, який надає можливість встановлення та налаштування Zabbix на серверах, які управляються Zabbix.

Більшість варіантів інсталяції Zabbix, зазвичай, є безкоштовними або мають безкоштовні версії. Проте, є деякі відмінності в ліцензуванні та доступних функціях в залежності від обраного методу та варіанту використання:

- **Відкрита версія (Open Source)**: Зазвичай, встановлення Zabbix з використанням офіційних пакунків або ручним способом використовує відкриту версію Zabbix, яка є безкоштовною і має відкритий вихідний код. Це означає, що ви можете використовувати та модифікувати програмне забезпечення безкоштовно, але деякі розширені функції можуть бути доступні тільки в комерційних версіях.
- **Комерційні версії**: Zabbix також пропонує комерційні версії з додатковими функціями, підтримкою та послугами, такими як розширена моніторингова функціональність, технічна підтримка, консультування, навчання та інші. Вартість цих версій може варіюватися в залежності від обсягу функціональності та рівня підтримки.
- **Хмарні рішення**: Хмарні рішення можуть мати різні цінові плани, від безкоштовних пробних версій до платних підписок. Ціни можуть залежати від обраного рівня функціональності, обсягу моніторингу, кількості ресурсів, що моніторяться та інших факторів.

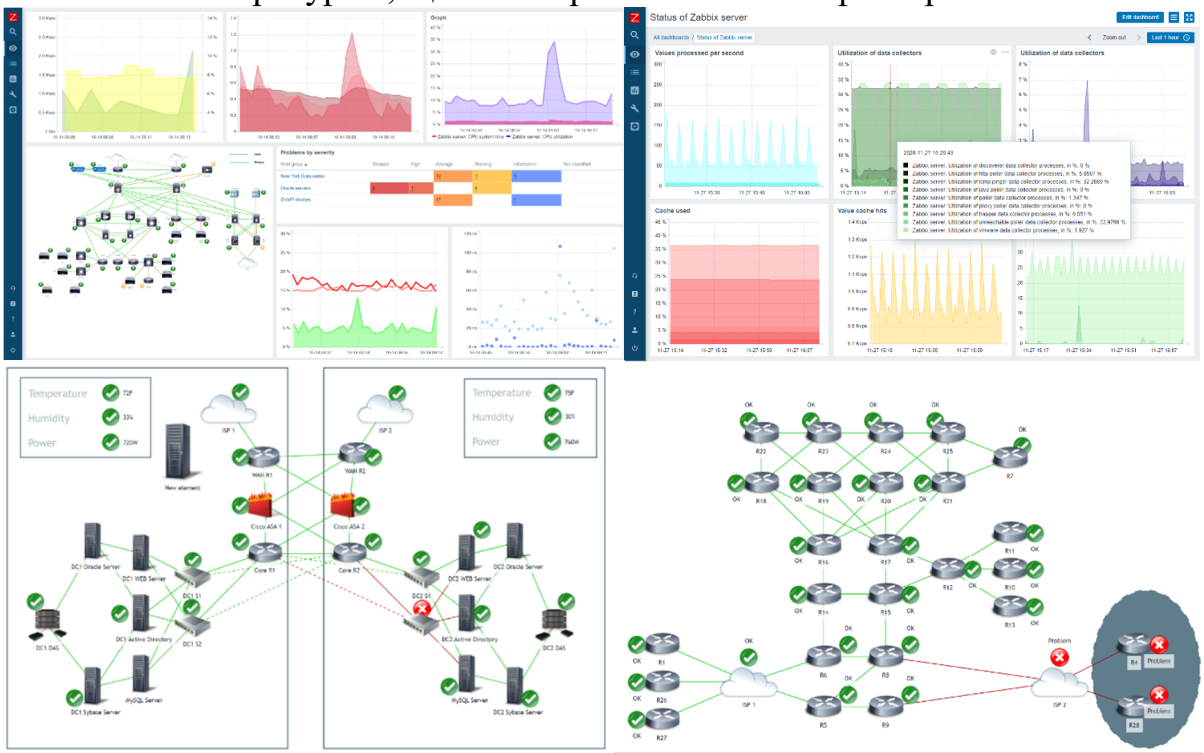


Рис. 5.3. Приклади візуалізації Zabbix

❖ **Потужна візуалізація даних.** Zabbix відображає зібрані дані різними можливими способами (рис.5.03). Можливо визначити панелі

інструментів на основі віджетів, що відображають необхідну інформацію:

- Великий вибір різноманітних віджетів
- Просте розміщення та масштабування віджетів за допомогою функції drag and drop
- Кожен віджет легко налаштовується відповідно до ваших потреб.
- Відображення метрик, проблем , інфраструктури та географічних карт на панелі управління
- Відображення поточної інформації про SLA бізнес-сервісів на панелі керування
- Дозволяє аналізувати та зіставляти показники за допомогою графіків, визначати користувацькі графіки.

❖ **Прив'язка об'єктів моніторингу до інтерактивної географічної карти** (рис.5.04).

- Вибір кількох постачальників геокарт
- Відображення географічного огляду вашого середовища на панелі інструментів Zabbix
- Доступ до будь-якого з об'єктів моніторингу з геокарти

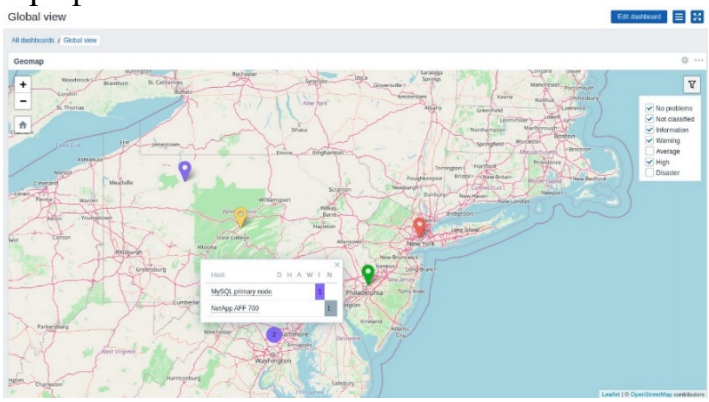


Рис. 5.4. Прив'язки об'єктів моніторингу Zabbix до інтерактивної карти.

- Угруповання об'єктів моніторингу у кластер на географічній карті
- Відстеження стану окремих об'єктів моніторингу чи всього кластера загалом

❖ **Проактивне реагування з прогнозуванням тенденцій.** Незважаючи на те, що для виявлення проблем добре мати пороги , ефективніше реагувати на проблеми проактивно. Функції прогнозування Zabbix допоможуть вам досягти цієї мети:

- Прогнозування значення для раннього оповіщення
- Прогнозування часу, що залишився до досягнення порогового значення проблеми

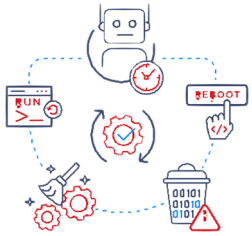
❖ **Отримання повідомлень про критичні проблеми**

- **Канали обміну повідомленнями.** Використовується кілька каналів обміну повідомленнями для оповіщення відповідальної особи або осіб про різні події, що відбуваються у середовищі:
  - Email
  - SMS для надійних оповіщень
  - Онлайн SMS-шлюзи
  - Платформи для спілкування :
    - ✓ Slack
    - ✓ MS Teams
    - ✓ Telegram
    - ✓ Express.ms
    - ✓ Rocket.chat
    - ✓ та багато інших…

  - Вебхуки для інтеграції із зовнішніми системами обміну повідомленнями, ITSM та системами тикетингу
- **Налаштування оповіщень.** Визначення різних повідомлень для різних каналів надсилання повідомлень. Можливо використовувати шаблони за замовчуванням або створити та налаштувати свій власний шаблон.
  - Налаштування повідомлення залежно від типу проблеми та ролі одержувача
  - Доповнення повідомлення будь-якою інформацією про час виконання та інвентар .
  - Надсилання запланованих PDF звітів для поглибленого та довгострокового аналізу даних

❖ **Zabbix може вирішувати деякі проблеми автоматично.**

За допомогою Zabbix можливо не тільки отримувати повідомлення про проблему, але й автоматично усувати її.



Для усунення проблеми може бути виконаний сценарій або команда:

- Перезапуск сервісу
- Керування хмарними ресурсами
- Виконання автоматичної зміни масштабу ресурсів
- Виконання будь-якої іншої логіки користувача

❖ **Використання єдиного інтерфейсу для всієї інфраструктури**

- Веб-інтерфейс Zabbix надає безліч способів представлення візуального огляду IT-середовища:

- Багатосторінкові панелі на основі віджетів
- Просте розміщення віджетів за допомогою функції drag and drop
- Налаштування інтервалів автоматичного оновлення панелі керування
- Можливість клонувати існуючу панель
- Приватні та публічні панелі

➢ Кожен із елементів панелі керування надзвичайно гнучкий, підтримує безліч режимів перегляду, фільтрів та враховує права доступу користувача:
- Фільтрування та відображення тільки необхідних даних
- Налаштування віджетів для відображення даних на різних рівнях деталізації
- При відображенні даних на панелі приладів дотримуються права доступу користувачів

➢ Збирання та відображення інформації про інвентаризацію
- Використання зібраних метрик для надання інвентаризаційної інформації про хости
- Поєднання вибірки власних даних інвентаризації з Zabbix API для надання додаткових даних інвентаризації
- Отримання огляду спільного інвентарю, групуванням хостів
- Надання та відстеження координат об'єктів моніторингу на геокарті
- Динамічне оновлення існуючої інформації про інвентар на основі зібраних метрик

❖ **Просте розгортання Zabbix у інфраструктурі**

➢ Моніторинг із коробки для провідних виробників програмного та апаратного забезпечення. Ось неповний перелік:
- Cisco
- HPE
- Microsoft
- IBM
- VMware
- Meraki
- Juniper
- F5

➢ Інтеграція Zabbix з існуючими системами.
- Моніторинг Docker контейнерів
- Web сервери - IIS, Apache, Nginx та інші
- Бази даних , такі як MySQL, PostgreSQL, Microsoft SQL, MongoDB та інші
- Моніторинг будь-якої операційної системи – Linux, Windows, Solaris, BSD, MacOS та інших
- Хмарні сервіси , такі як AWS, Amazon cloud, Google cloud та інші
- Сервіси IP-телефонії

3. Nagios vs Zabbix

Nagios® ***vs*** ZABBIX

Ті, хто давно працює у світі IT – вважають Nagios “стандартом де-факто” у сфері моніторингу з відкритим вихідним кодом. І це правда, тому що ця компанія, яка перша почала застосовувати моніторинг правильно. До Nagios існували програми, але вони були дуже дилетантськими. Так само існували окремі якісні інструменти, які були гарними лише для конкретного завдання. Перша версія Nagios датується минулим століттям: 1999 роком. Минуло багато років, і технологія розвивалася: Nagios розвивався через екосистему сторонніх доповнень, якими намагалися доповнити функції, що бракувало системі. Ці розробки розширили можливості налаштування Nagios та перетворили його на універсальний інструмент.

Zabbix з'явився у 2001 році. Це повноцінна розробка, а не відгалуження Nagios, і її головна особливість у тому, що вона має більш цілісний погляд на моніторинг, що охоплює продуктивність, а не тільки стан, оскільки це - один із найістотніших недоліків Nagios. Zabbix також має систему управління WEB, що дозволяє керувати ним централізовано, без громіздких конфігураційних файлів, як це було у випадку з Nagios.

Zabbix та Nagios вимагають встановлення безлічі плагінів, щоб бути ефективними та пропонувати повний набір функціональних можливостей. Zabbix не має «офіційної» бібліотеки плагінів для спільноти, хоча має список OID для SNMP-запитів. Крім того, він не пропонує можливості роботи з інструментами Enterprise, такими як Oracle, Exchange, Active Directory та іншими ядрами, на відміну від Nagios, який завдяки своїм розширенням це «вміє».

Nagios має на 100% відкриту гігантську бібліотеку, але за більшістю складових цієї бібліотеки немає компанії, що її підтримує, або дбає про її актуальність. Це один з його основних недоліків.

Що стосується спільноти користувачів, то знову, найбільшою спільнотою є Nagios, з тієї простої причини, що він з'явився першим. Щодо кількості відгалужень та форків Nagios вже згадувалось у цій лекції. Велика кількість відгалужень Nagios призводить до деякої хаотичності екосистеми, коли доходить до впровадження плагінів або інструментів під одну задачу, розроблених різними розробниками. Кожна гілка має свою філософію і згодом робить її сумісність з іншими гілками та з батьківським проектом (Nagios) «через танці з бубном». Але також треба відмітити, що така різноманітність це палиця з двома кінцями. Вона дозволяє знайти плагін під потреби користувача, а не використовувати лише те, що прийшло «з коробки».

4. Огляд Prometeus

- **Що таке Prometheus.** Це набір інструментів для моніторингу та попередження систем із відкритим вихідним кодом, створений у SoundCloud. З моменту створення в 2012 році багато компаній і організацій впровадили Prometheus, і проект має активну спільноту розробників і користувачів . На сьогодні, це окремий проект із відкритим кодом, який підтримується незалежно від будь-якої компанії. Щоб підкреслити це та уточнити структуру управління проектом, Prometheus приєднався до Cloud Native Computing Foundation у 2016 році як другий розміщений проект після Kubernetes.

  Prometheus збирає та зберігає свої показники як дані часових рядів, тобто інформація про показники зберігається з міткою часу, коли вона була записана [48], поряд із необов'язковими парами ключ-значення, які називаються мітками.

- **Основні риси Prometheus.**
  - **Багатовимірна модель даних із даними часових рядів, ідентифікованих назвою метрики та парами ключ/значення.**
    До речі, про цю принципову особливість Prometheus ми згадували у другій лекції. Prometheus принципово зберігає всі дані як часові ряди: потоки значень із мітками часу, що належать до того самого показника та того самого набору мічених параметрів. Окрім збережених часових рядів, Prometheus може генерувати тимчасові похідні часові ряди в результаті запитів.
  - **PromQL, гнучка мова запитів для використання цієї розмірності.** PromQL (Prometheus Query Language) дозволяє користувачеві вибирати та агрегувати дані часових рядів у реальному часі [48]. Результат виразу можна або відобразити у вигляді графіка, переглянути як табличні дані в браузері виразів Prometheus, або використати зовнішніми системами через HTTP API .
  - **Відсутність залежності від розподіленого сховища**; окремі серверні вузли є автономними
  - **Pull model.** Збирання часових рядів відбувається через модель вилучення через HTTP. Prometheus сама відправляє запити на різні джерела даних і отримує від них потрібну інформацію. Це може бути, наприклад, інформація про стан різних систем або програм, що моніторяться. Основна перевага такого підходу полягає в тому, що Prometheus може ефективно збирати дані з різних джерел, що дозволяє моніторити різні системи або сервіси у вашому середовищі. Крім того, цей підхід надає гнучкість у налаштуванні та спрощує процес системного моніторингу, оскільки ви можете легко налаштувати, які дані ви хочете збирати та звідки [48].

- **Prometheus Pushgateway.** Проштовхування часових рядів підтримується через проміжний шлюз. Час від часу Prometheus потрібно стежити за компонентами, які неможливо очистити. Prometheus Pushgateway дає змогу переміщати часові ряди з короткотривалих пакетних завдань на рівні обслуговування до проміжного завдання, яке Prometheus може вилучити [17]. У поєднанні з простим текстовим форматом експозиції Prometheus це дозволяє легко інструментувати навіть сценарії оболонки без клієнтської бібліотеки.
- **Цілі виявляються за допомогою виявлення служби або статичної конфігурації.** Існує два основних методи виявлення цілей у Prometheus **Service Discovery** та **статична конфігурація**:
  - **Виявлення служби (Service Discovery).** Prometheus може використовувати вбудовані механізми виявлення служб для автоматичного визначення цілей [58]. **Service Discovery** в Prometheus полягає у виявленні автоматично цілей моніторингу - це можуть бути різні системи, сервіси або компоненти, які ви хочете включити до моніторингу. Prometheus може використовувати вбудовані механізми виявлення служб для цього. Основні методи виявлення служб включають:
    1. **DNS виявлення**. Prometheus може використовувати DNS для автоматичного виявлення служб. Вказуються шаблони імен хостів, які використовуються для моніторингу, і Prometheus автоматично виявить ці хости.
    2. **Consul або Kubernetes**. В середовищі, де використовуються інструменти, такі як Consul (інструмент для розподіленої системи конфігурації, виявлення служб та управління цими службами) або Kubernetes (система оркестрації контейнерів, що призначена для автоматизації розгортання, масштабування та керування додатками, що розгорнуті у контейнерах), Prometheus може використовувати їхні API для автоматичного виявлення цілей моніторингу. Наприклад, в Kubernetes Prometheus може автоматично виявляти всі сервіси, що запущені у кластері.
    3. **EC2 Service Discovery**. В середовищі AWS, Prometheus може автоматично виявляти цілі моніторингу, які базуються на екземплярах EC2, використовуючи дані, які надаються сервісом EC2 та іншими AWS-специфічними інструментами.
    4. **Azure Service Discovery**. Аналогічно, для середовищ Azure існують механізми виявлення, які можуть бути використані для автоматичного визначення цілей моніторингу на базі інформації про ресурси та служби, що надаються платформою Azure.

5. **GCP Service Discovery**. Google Cloud Platform (GCP) також має вбудовані інструменти для виявлення служб, які можуть бути використані Prometheus для автоматичного визначення цілей моніторингу в середовищі GCP.
6. **OpenStack Service Discovery**. В середовищі OpenStack, існують різні інструменти та API, які можуть бути використані для автоматичного виявлення служб і ресурсів для моніторингу.

Перелічені механізми виявлення служб дозволяють автоматизувати процес додавання та видалення цілей моніторингу, що спрощує управління системою моніторингу, особливо в великих та динамічних середовищах.

- **Статична конфігурація.** Користувачі можуть вказати цілі (адреси систем або сервісів, де розташовані експортери) явно в конфігураційному файлі prometheus.yml. Цей підхід використовується для статичного визначення цілей без виявлення служби, що може бути корисним у випадках, коли сервери відомі заздалегідь, або коли потрібна точна конфігурація лише обраних хостів.

➢ **Підтримка кількох режимів побудови графіків і інформаційних панелей.** Два найпоширеніших інструменти для цього - Grafana та Prometheus's own built-in Expression Browser.

- **Grafana** - це інструмент візуалізації та моніторингу, який може використовувати дані, зібрані Prometheus. Він надає широкий спектр можливостей для створення графіків, інформаційних панелей та панелей моніторингу за допомогою різних типів даних, функцій та налаштувань візуалізації. Grafana дозволяє створювати діаграми, графіки, теплові карти, стовпчаті та кругові діаграми та інші типи візуалізацій для аналізу та відображення даних, що збираються Prometheus.

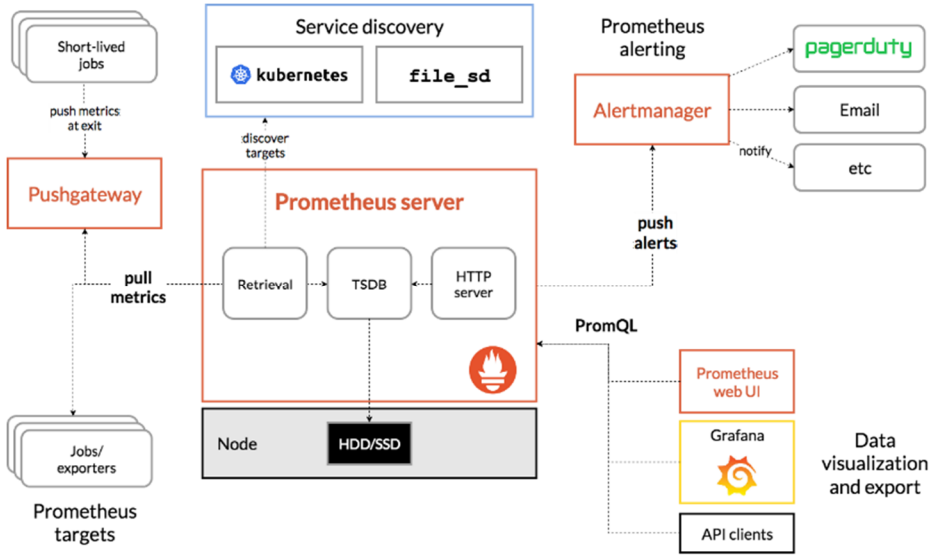


Рис. 5.05. Архітектура Prometheus і деякі компоненти його екосистеми.

- Вбудований **Expression Browser** в Prometheus надає можливість візуалізувати дані непрямо в інтерфейсі самого Prometheus. Він використовується для відображення графіків та відслідковування значень метрик прямо в браузері. Це зручний спосіб швидко переглянути дані та здійснити базовий аналіз без необхідності налаштування додаткових інструментів візуалізації.

❖ **Архітектура Prometheus.** Екосистема Prometheus складається з кількох компонентів (рис.5.05), багато з яких є необов'язковими.

- **Головний сервер Prometheus** , який збирає та зберігає дані часових рядів
- **Клієнтські бібліотеки** для інструментування програмного коду
- **Push-шлюз** для підтримки короткострокових завдань
- **Експортери** спеціального призначення для таких сервісів, як HAProxy, StatsD, Graphite тощо.
- **Менеджер сповіщень** для обробки сповіщень
- **Засоби підтримки**

Більшість компонентів Prometheus написані на Go , що полегшує їх створення та розгортання як статичних двійкових файлів.

Prometheus збирає показники з інструментальних завдань безпосередньо або через проміжний push-шлюз для короткострокових завдань. Він зберігає всі зібрані зразки локально та запускає правила для цих даних, щоб агрегувати та записувати нові часові ряди з наявних даних або створювати сповіщення. Для візуалізації зібраних даних можна використовувати Grafana або інші споживачі API.

Незважаючи на те, що Prometheus можна використовувати для побудови графіків конкретних запитів, це не повноцінна інформаційна панель і її потрібно підключити до Grafana для створення інформаційних панелей. Це є суттєвим недоліком системи через додаткову складність налаштування.

❖ Докладніше про схему взаємодії компонентів системи моніторингу з урахуванням Prometheus. Базова конфігурація складається з трьох компонентів екосистеми (рис.5.06):

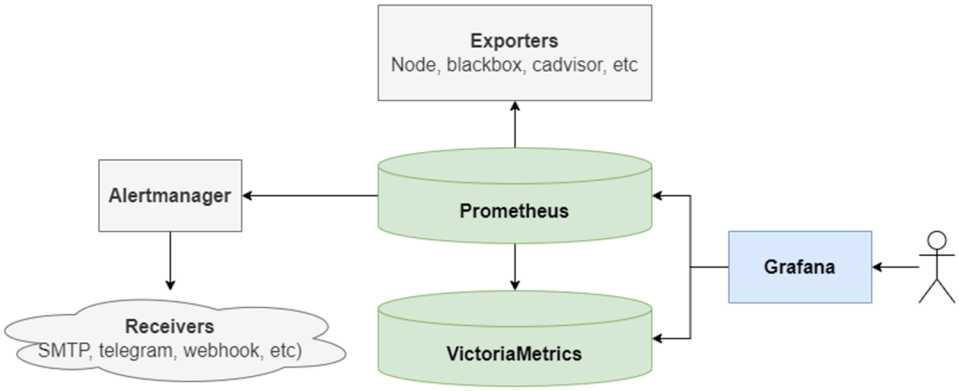


Рис. 5.06. Схема взаємодії компонентів Prometheus (VictoriaMetrics).

- **Експортери (exporters).** Експортер збирає дані та повертає їх у вигляді набору метрик. Експортери поділяються на офіційні (написані командою Prometheus) та неофіційні (написані розробниками різного програмного забезпечення для інтеграції з Prometheus). За необхідності є можливість писати свої експортери та розширювати наявні додатковими метриками.
- **Prometheus.** Отримує метрики від експортерів і зберігає їх у БД часових рядів. Підтримує потужну мову запитів PromQL (Prometheus Query Language) для вибірки та агрегації метрик. Дозволяє будувати прості графіки та формувати правила повідомлень (alerts) на основі виразів PromQL для відправки через Alertmanager
- **Alertmanager.** Обробляє повідомлення від Prometheus і розсилає їх. За допомогою механізму приймачів (receivers) реалізовано інтеграцію з поштою (SMTP), Telegram, Slack та ін. системами, а також надсилання повідомлень у власний API за допомогою вебхуків (webhook). Таким чином, базова конфігурація дозволяє збирати дані, писати складні запити та надсилати повідомлення на їх основі [48]. Однак, по-справжньому потенціал Prometheus розкривається при додаванні двох додаткових компонентів (або як мінімум одного – Grafana)
- **VictoriaMetrics.** Отримує метрики з Prometheus за допомогою remote write [11]. Підтримує мову запитів MetricsQL , синтаксис якого сумісний із PromQL. Надає оптимізоване споживання ресурсів зберігання даних і високопродуктивне виконання запитів. Ідеально підходить для довготривалого зберігання великої кількості метрик
  - **Cortex** — аналогічне рішення (з іншою архітектурою) [13].
  - **Mimir** (від Grafana Labs) — фактично форк Cortex, оптимізований [38].
  - **Thanos** — перетворює Prometheus у розподілену систему моніторингу з можливістю централізованого збору, збереження й аналізу метрик на великих кластерах [8].
- **Grafana.** Надає засоби візуалізації та додаткового аналізу інформації з Prometheus та VictoriaMetrics. Є приклади дашбордів практично під

будь-які завдання, які за потреби можна легко доопрацювати. Створення власних дашбордів також інтуїтивно (зрозуміло, крім деяких тонкощів) – достатньо знати основи PromQL / MetricsQL Де-факто використання Grafana разом з Prometheus вже стало стандартом, у той час як додавання в конфігурацію VictoriaMetrics безумовно опціональне і необхідне для високонавантажених систем.

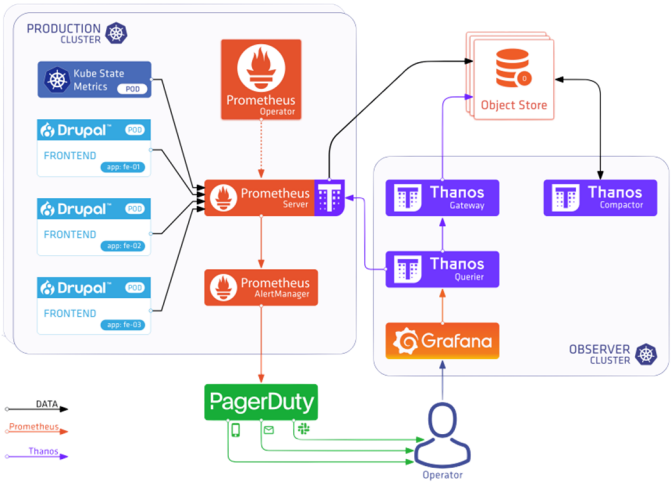


Рис. 5.07. Схема взаємодії компонентів Prometheus (багатокластерний моніторинг Thanos).

Порівняльна таблиця: Thanos (рис.5.07)**, VictoriaMetrics, Cortex та Grafana Mimir**

Критерій	**Thanos**	**VictoriaMetrics**	**Cortex**	**Grafana Mimir**
Архітектура	Надбудова над Prometheus; мікросервісна архітектура з компонентами: Sidecar, Store, Querier, Compactor, Receive.	Самостійна time-series база; доступна в режимах single-node та кластер.	Мікросервісна архітектура; компоненти: Ingester, Distributor, Querier, Store Gateway, Compactor.	Форк Cortex; мікросервісна архітектура з можливістю монолітного режиму; компоненти схожі на Cortex
Джерело даних	Працює зі стандартними Prometheus TSDB файлами; інтеграція через Sidecar або Receive	Підтримує Prometheus remote_write, Influx, Graphite, OpenTSDB API, native vmagent.	Підтримує Prometheus remote_write; інтеграція через Distributor.	Підтримує Prometheus remote_write; інтеграція через Distributor; додаткові оптимізації

	для прийому метрик.			для високої продуктивності
Зберіган-ня	Потребує зовнішнього об’єктного сховища (S3, GCS, MinIO, Azure Blob).	Локальне (SSD/HDD) або мережне сховище; не потребує об’єктного сховища.	Потребує об’єктного сховища для довготривалого зберігання; підтримує S3, GCS та інші.	Потребує об’єктного сховища; підтримує S3-подібні сховища; оптимізоване зберігання для високої масштабованості
Масшта-бування	Горизонтальне через додавання компонентів; кожен компонент можна масштабувати окремо.	Вбудоване; кластерний режим дозволяє масштабувати окремі вузли для зберігання, запису та запитів.	Горизонтальне масштабування через додавання інстансів компонентів; висока масштабованість для великих навантажень	Горизонтальне масштабування; додаткові оптимізації для покращення продуктивності та зменшення затримок.
Довго-тривале зберіган-ня	Дані архівуються у об’єктне сховище після певного періоду; Store Gateway забезпечує доступ до історичних даних.	Вбудоване довготривале зберігання; підтримує retention-політики; не потребує зовнішнього об’єктного сховища.	Дані зберігаються в об’єктному сховищі; підтримує retention-політики; забезпечує ефективний доступ до історичних даних.	Дані зберігаються в об’єктному сховищі; оптимізоване зберігання для швидкого доступу та обробки великих обсягів даних
Висока доступ-ність (HA)	Підтримка HA для Prometheus і Querier; кожен	Вбудована HA в кластерній версії; автоматичне	Вбудована HA; дані реплікуються між	Вбудована HA; додаткові оптимізації

	Prometheus інстанс потрібно дублювати для HA.	відновлення при збоях вузлів	інстансами Ingester для забезпечення доступності.	для швидшого відновлення та мінімізації втрат даних при збоях
Push-метрики	Підтримує через Thanos Receive (альтернатива remote_write); дозволяє напряму пушити метрики	Пушить метрики через vmagent; підтримує Prometheus remote_write та інші формати.	Підтримує Prometheus remote_write для пушу метрик; інтеграція через Distributor.	Підтримує Prometheus remote_write; оптимізований прийом метрик для високих навантажень
Прозорість PromQL	Використовує оригінальний PromQL; 100% сумісність з Prometheus	Підтримує більшість PromQL-запитів; є деякі обмеження та розширення.	Повна підтримка PromQL; забезпечує сумісність з існуючими дашбордами та запитами	Повна підтримка PromQL; додаткові оптимізації для швидшого виконання запитів
Складність розгортання	Висока; потребує налаштування кількох компонентів та інтеграції з об'єктним сховищем.	Відносно проста; один бінарний файл для single-node; кластерний режим потребує додаткового налаштування.	Середня; потребує налаштування кількох компонентів; інтеграція з об'єктним сховищем.	Середня; підтримує монолітний режим для спрощеного розгортання; кластерний режим потребує налаштування кількох компонентів.

Рекомендації щодо вибору:

- ✓ **Thanos**: підходить, якщо вже використовуєте Prometheus і потребуєте довготривалого зберігання та високої доступності; готові до налаштування кількох компонентів та інтеграції з об'єктним сховищем.

- ✓ **VictoriaMetrics**: оптимальний вибір для швидкого розгортання з мінімальними ресурсами; підходить для сценаріїв з високими вимогами до продуктивності та простоти налаштування.
- ✓ **Cortex** та **Grafana Mimir**: підходять для великих масштабованих серед

❖ **Інсталяція Prometheus.**

- ➢ З використанням **офіційних пакунків** для різних операційних систем, таких як Linux, Windows та macOS. Ви можете завантажити ці пакунки з веб-сайту Prometheus та встановити їх за допомогою стандартних процедур для вашої операційної системи.
- ➢ **Встановлення** Prometheus **у контейнері Docker** є популярним методом, оскільки це дозволяє легко запускати та масштабувати Prometheus на різних платформах. Ви можете завантажити образ Prometheus з Docker Hub і запустити контейнер за допомогою команд Docker.
- ➢ Prometheus може бути розгорнута **в середовищі Kubernetes як контейнеризований додаток**. Це дозволяє автоматизувати розгортання, масштабування та управління Prometheus за допомогою інструментів Kubernetes.
- ➢ Як альтернативу, ви можете встановити **Prometheus вручну**, завантаживши вихідний код з GitHub та компілюючи його на своєму сервері. Цей метод може бути корисним для більш гнучкого налаштування або для встановлення Prometheus на платформи, для яких немає офіційних пакунків.

Платформи, на яких можна встановити Prometheus, включають усі основні операційні системи, такі як Linux, Windows та macOS. Крім того, вона також може бути встановлена на хмарних платформах, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure, а також на власних серверах або віртуальних машинах.

❖ **Коли підходить?**

Prometheus добре працює для запису будь-яких чисто числових часових рядів. Він підходить як для машиноцентричного моніторингу, так і для моніторингу високодинамічних сервіс-орієнтованих архітектур. У світі мікросервісів його підтримка збору багатовимірних даних і запитів є особливою перевагою.

Prometheus розроблено для надійності, щоб бути системою, до якої ви звертаєтеся під час збою, щоб дозволити вам швидко діагностувати проблеми. Кожен сервер Prometheus є автономним і не залежить від мережного сховища чи інших віддалених служб. На цю систему моніторингу можете повністю покластися, коли інші частини вашої інфраструктури зламано, і не потрібно налаштовувати розгалужену інфраструктуру, щоб використовувати його.

❖ **Коли не підходить?**

Prometheus цінує надійність. Завжди можливо переглянути доступну статистику щодо системи, навіть за умов збою. Якщо необхідна 100% точність, наприклад для виставлення рахунків за запитом, Prometheus не є гарним вибором, оскільки зібрані дані, швидше за все, не будуть достатньо детальними та повними. У такому випадку краще використовувати якусь іншу систему для збору й аналізу даних для виставлення рахунків.

5. Grafana

Вже кілька разів згадувалась назва продукту Grafana [59]. Grafana — потужна система для візуалізації та моніторингу даних, яка підтримує різні джерела даних. Вона дозволяє створювати дашборди, графіки, таблиці, а також налаштовувати алерти на основі зібраних метрик. Grafana є окремим продуктом, який може працювати з багатьма системами збору даних. Наприклад ми тільки що бачити типову схему побудови системи моніторингу Prometheus з візуалізацією даних за допомогою Graphana. Крім того є довгий перелік моніторингових систем, у які Graphana успішно інтегрується:

- ✓ **InfluxDB** – для збереження часових рядів.
- ✓ **Elasticsearch** – для аналізу логів.
- ✓ **Loki** – система збору логів від Grafana Labs.
- ✓ **Zabbix** – через спеціальний плагін.
- ✓ **Nagios** – можливе підключення через API або додаткові плагіни.
- ✓ **MySQL, PostgreSQL** – для роботи з бізнес-даними.
- ✓ **Azure Monitor, AWS CloudWatch** – для моніторингу хмарних ресурсів.

Grafana не є повноцінною системою збору метрик, тобто вона сама не виконує моніторинг у традиційному розумінні (не збирає метрики безпосередньо з серверів, мережного обладнання чи додатків). Проте останні версії Grafana (зокрема Grafana Cloud або Grafana Agent) мають базові можливості збору метрик та їх передачі в підтримувані системи моніторингу.

Самостійно Grafana може:

- ✓ Збирати власні метрики продуктивності (завантаження CPU, використання пам’яті, запити до бази даних).
- ✓ Моніторити доступність сервісів через плагін Grafana Synthetic Monitoring (перевірка HTTP/HTTPS-запитів, Ping).
- ✓ Збирати логи та події через Grafana Loki (аналог Elastic Stack для логів).

- ✓ Збирати прості системні метрики через Grafana Agent (CPU, RAM, Disk, Network), але він працює лише як колектор для Prometheus або інших сховищ даних.

Grafana сама по собі безагентна, бо працює з уже зібраними метриками. Але якщо використовувати Grafana Agent, то це вже агентний підхід, бо агент встановлюється на сервери та передає дані.

Grafana в основному візуалізує часові ряди (time-series data), тому її головний фокус — динамічні метрики, такі як:

- ✓ Інфраструктурні метрики (CPU, RAM, Disk I/O, Network).
- ✓ Метрики продуктивності додатків (відгук API, кількість запитів, помилки, час виконання).
- ✓ Метрики кібербезпеки (кількість атак, блокованих запитів, аномальна активність).
- ✓ Бізнес-метрики (прибуток, трафік користувачів, конверсії).

Історія створення та розвитку Grafana

У 2013 році вийшла Kibana 3, що була частиною ELK-стеку (Elasticsearch, Logstash, Kibana) і використовувалася для візуалізації логів та аналітики даних. Kabana 3 дозволяла будувати дашборди, графіки та таблиці для аналізу логів, фільтрувати та виконувати пошук по логах у реальному часі на основі даних з Elasticsearch.

Початок (2014)

Grafana була створена Токе Йонсеном (Torkel Ödegaard) як форк Kibana 3, але з акцентом на візуалізацію часових рядів замість аналізу логів. Спочатку Grafana працювала лише з Graphite, популярним у той час сховищем метрик. Вона швидко отримала популярність серед DevOps-спільноти завдяки можливості будувати зручні дашборди.

Ранні версії **(2015-2017)** – підтримка нових джерел даних

Grafana 2.0 (2015) – додано підтримку InfluxDB, OpenTSDB, Prometheus, що зробило Grafana універсальним інструментом для моніторингу.

Grafana 3.0 (2016) – з'явилася нова система плагінів, що дозволило додавати нові джерела даних та панелі візуалізації.

Grafana 4.0 (2016) – введено систему алертів, що дало можливість отримувати сповіщення про аномалії.

Розширення можливостей **(2018-2020)** – робота з логами та бізнес-аналітика

Grafana 5.0 (2018) – новий інтерфейс, підтримка папок та оновлений UI для дашбордів.

Grafana 6.0 (2019) – інтеграція з Loki (система збору логів від Grafana Labs), що зробило платформу конкурентом ELK Stack.

Grafana 7.0 (2020) – API для data transformation, що дозволяє редагувати метрики перед візуалізацією.

Останні версії (2021-2024) – Machine Learning та розширена безпека

Grafana 8.0 (2021) – покращені алерти, інтеграція з machine learning-моделями для прогнозування аномалій.

Grafana 9.0 (2022) – додано Grafana OnCall для спрощеного керування інцидентами.

Grafana 10.0 (2023) – підтримка нових хмарних сервісів, покращена робота з базами даних, покращення продуктивності.

Зараз Grafana є одним із найпопулярніших інструментів моніторингу, який використовується у DevOps, кібербезпеці та бізнес-аналітиці.

Операційні системи, на яких працює Grafana

Grafana є кросплатформеним застосунком і підтримує такі операційні системи:

✓ Linux (Debian, Ubuntu, CentOS, RHEL, Fedora, SUSE)

✓ Windows

✓ macOS

✓ Docker (як контейнер)

✓ Kubernetes (через Helm Charts)

На момент написання посібника доступні такі способи встановлення:

1. **Інсталяція з офіційних пакетів.**
   Linux (Debian/Ubuntu)
   wget https://dl.grafana.com/oss/release/grafana_10.3.1_amd64.deb
   sudo dpkg -i grafana_10.3.1_amd64.deb
   sudo systemctl start grafana-server
   Linux (CentOS/RHEL)
   sudo yum install -y https://dl.grafana.com/oss/release/grafana-10.3.1-1.x86_64.rpm
   sudo systemctl start grafana-server
   Windows – через завантаження .zip-архіву з офіційного сайту, розпакування та запуск grafana-server.exe.
   macOS
   brew install grafana
   brew services start grafana
2. **Розгортання в Docker**
   Один із найзручніших способів для тестування та продакшену.
   docker run -d -p 3000:3000 --name=grafana grafana/grafana
   Якщо потрібно зберігати дані:

***docker run -d -p 3000:3000 --name=grafana ***
***-v grafana-storage:/var/lib/grafana ***
grafana/grafana

3. **Розгортання в Kubernetes**
   Для Kubernetes рекомендується використовувати Helm-чарт:
   helm repo add grafana https://grafana.github.io/helm-charts
   helm install my-grafana grafana/grafana
4. **Хмарні рішення (SaaS)**
   Якщо не хочеться розгортати Grafana вручну, можна використати Grafana Cloud, де є безкоштовний тариф.

Таким чином, Grafana – це потужний та гнучкий інструмент для візуалізації даних, який вийшов за межі класичного моніторингу. Завдяки своїй кросплатформенності, підтримці численних джерел даних і розширюваності, вона стала не лише стандартом у сфері IT-операцій та моніторингу, а й потужним засобом для бізнес-аналітики та дослідження даних.

6. Практичне використання інструментів для моніторингу

- **Налаштування моніторингу ресурсів серверів.** Моніторинг ресурсів серверів є критичним аспектом у забезпеченні надійності та ефективності IT-інфраструктури. У цьому контексті, розглянемо процес налаштування моніторингу для основних ресурсів сервера: центральний процесор (CPU), оперативна пам'ять (RAM), пристрої зберігання (диск), та мережні підключення.
  - **Моніторинг CPU, пам'яті, диска та мережі**
    - **CPU.** Використовуйте моніторингові інструменти або агенти для вимірювання використання CPU. Метрики можуть включати загальне використання, завантаженість окремих ядер та інші параметри. Встановлення порогів для використання CPU має виконуватися з огляду на стандарт «заліза».
      Наприклад, алерт при використанні CPU, що перевищує 90% протягом певного часу (30-40 хв) для хмарних хостів, 85% до 60 хв для віртуальних хостів та 80% до 30 хв для дискретного сервера.
    - **RAM.** Моніторте використання оперативної пам'яті для уникнення переповнення. Зокрема, стежте за вільною та використаною RAM. Встановлення порогів для використання оперативної пам'яті.
      Наприклад, алерт при використанні RAM більше 80%.
    - **Диск.** Моніторте доступність простору на диску, швидкість читання/запису та ступінь використання файлових систем. Встановлення порогів для вільного простору на диску та швидкості

читання/запису. Наприклад, алерт при вільному просторі менше 10% логічного диску.

- ❖ **Мережа.** Вимірюйте пропускну здатність та використання мережі. Встановлення порогів для використання пропускної здатності. Наприклад, алерт при використанні мережі більше 90%.

- ➢ **Визначення порогів та алертів**
  - ❖ **Визначення порогів** повинно відбуватися відповідно до конкретних потреб інфраструктури та додатків. Різні сценарії можуть вимагати різних значень порогів.
  - ❖ **Тестування рішення.** Перевірка обраного рішення для визначення, які пороги є оптимальними для системи. Тестування дозволяє виявити можливі проблеми та уточнити налаштування, яке має зменшити хибні спрацювання алертів, але зберегти високу функціональність моніторингу.
- ➢ **Алерти**
  - ❖ **Конфігурація системи алертів.** Визначення того, як система повинна відправляти алерти - електронною поштою, через систему керування подіями, або іншими засобами. Як правило, конфігурація алертів прописується у стандартах підприємства. Є корпоративні канали спілкування, які і обираються для відправки таких повідомлень.
  - ❖ **Задання пріоритетів.** Розробка системи пріоритетів для алертів. Наприклад, алерти про високе використання CPU можуть мати вищий пріоритет, ніж алерти про вільне місце на диску.
  - ❖ **Визначення дій**. Визначення конкретних дій, які слід приймати при отриманні алерту. Це може включати автоматичні заходи, збір інформації для аналізу або відправку повідомлення адміністратору.

Налаштування моніторингу ресурсів серверів включає в себе не лише визначення того, що слід моніторити, але і як ефективно використовувати алерти для вчасного виявлення проблем і забезпечення оптимального функціонування інфраструктури.

- **Моніторинг додатків.** Розглянемо процес запуску та конфігурації моніторингу для додатків, а також важливість відслідковування продуктивності та доступності.
  - ➢ **Запуск та конфігурація моніторингу додатків**
    - ❖ **Вибір моніторингових інструментів**, а саме визначення, які інструменти моніторингу будуть використовуватися. Деякі додатки можуть мати вбудовані засоби моніторингу, або ви можете використовувати універсальні рішення, такі як Prometheus, Nagios, Zabbix чи інші.

- ❖ **Конфігурація моніторингових агентів.** Якщо додаток вимагає агент для моніторингу, налаштуйте агенти на серверах, де розгорнуто додаток.
- ❖ **Визначення параметрів моніторингу**. Оберіть, які параметри додатку слід моніторити. Це може включати використання ресурсів (CPU, пам'ять, диск), стан служб, пропускну здатність мережі та інші ключові метрики.
- ❖ **Налаштування порогів та алертів.** Встановлення порогів для параметрів моніторингу та алертів. Наприклад, алерт при перевищенні певного рівня використання ресурсів або при відключенні служби.

➢ **Відслідковування продуктивності та доступності**

- ❖ **Використання ресурсів.** Спостерігайте за використанням CPU, оперативної пам'яті, диска та інших ресурсів, щоб вчасно виявляти проблеми та забезпечити оптимальну продуктивність.
- ❖ **Логи та події.** Аналізуйте логи додатку для виявлення помилок, попереджень та інших важливих подій, які можуть вказувати на проблеми у роботі додатку.
- ❖ **Моніторинг відгуків**. Вимірюйте час відгуку додатку для оцінки його продуктивності та відчуття кінцевого користувача та спостерігайте доступність додатку за допомогою пінгів або HTTP-запитів.
- ❖ **Алерти та автоматизація відновлення.** Встановіть алерти для виявлення проблем доступності та автоматизуйте відновлення сервісів або додатку в разі виявлення неполадок.
- ❖ **Тестування відмовостійкості.** Проводьте тестування відмовостійкості для переконання в тому, що додаток може витримати можливі неполадки та продовжувати працювати.

Ефективний моніторинг додатків дозволяє оперативно реагувати на проблеми, підтримувати високу доступність та оптимізувати продуктивність.

7. Налаштування сповіщень та автоматизація управління системним моніторингом

- **Встановлення правил сповіщень.** Розглянемо, як правильно вибирати сповіщення для різних сценаріїв та налаштовувати ескалації.

➢ **Вибір правильних сповіщень для різних сценаріїв**

- ❖ **Критичні алерти.** Використовуйте SMS або push-сповіщення для найважливіших алертів. Електронні листи можуть бути ефективними лише для певних категорій критичних подій.
- ❖ **Загроза безпеки або інциденти.** Використовуйте телефонні дзвінки або системи голосового сповіщення для оперативного реагування на загрози безпеки. SMS або електронні листи можуть

додатково використовуватися для швидкого розповсюдження інформації.

- ❖ **Інформаційні повідомлення.** Електронні листи чи сповіщення через платформи спільної роботи підходять для інформаційних повідомлень, які не потребують негайного реагування.
- ❖ **Врахування ролей та відповідальностей.** Налаштуйте сповіщення відповідно до ролей користувачів і їх відповідальностей. Наприклад, інженерам можуть надсилатися технічні деталі алертів, тоді як менеджерам - загальна інформація.
- ❖ **Групові сповіщення.** Використовуйте сповіщення для груп користувачів, коли проблема впливає на кілька відділень або команд.

➢ **Конфігурація ескалацій**

- ❖ **Визначення тайм-фреймів.** Налаштуйте часові інтервали для ескалацій, щоб забезпечити, що проблеми будуть вирішені у визначені строки.
- ❖ **Автоматизовані ескалації.** Використовуйте автоматизовані системи для ескалацій, щоб уникнути затримок у випадках, коли персонал не реагує.
- ❖ **Особистий зв'язок.** Встановлюйте правила для особистого зв'язку, таких як телефонні дзвінки, в разі важливих або неурядових подій.
- ❖ **Включення вищого керівництва.** Ескалюйте алерти до вищого керівництва у випадках, коли проблема вимагає внутрішньоорганізаційних рішень чи рішень на вищому рівні.
- ❖ **Журналювання подій.** Ведення журналу подій та аналіз ескалацій для подальшого вдосконалення процесу моніторингу.
- ❖ **Оцінка ефективності.** Регулярно оцінюйте ефективність ескалацій та внесіть зміни в процеси, якщо це необхідно.

Встановлення правил сповіщень та ескалацій важливо для забезпечення того, щоб відповідальні особи отримували необхідну інформацію у вчасний спосіб та вчасно реагували на події, що виникають в інфраструктурі.

- **Автоматизація реакції на події.** Розглянемо використання скриптів та автоматизаційних засобів, а також інтеграцію з іншими системами управління.

➢ **Використання скриптів та автоматизаційних засобів**

- ❖ **Розробка автоматизованих сценаріїв.** Використовуйте мови програмування або спеціальні скриптові мови для розробки сценаріїв автоматизації.
- ❖ **Взаємодія з API**. Використовуйте API для взаємодії з іншими системами та автоматизації завдань.

- ❖ **Використання Ansible, Puppet, Chef тощо.** Інтегруйте засоби автоматизації конфігурації та деплойменту для автоматизації відновлення стану системи.
- ❖ **Системи оркестрації, такі як Kubernetes.** Використовуйте для автоматизації управління контейнерами та додатками.

➢ **Інтеграція з іншими системами управління**

- ❖ **API і інтеграційні рішення.** Використовуйте API для взаємодії між різними інструментами моніторингу та управління.
- ❖ **Інтеграція з системами керування подіями (SIEM).** Передавайте дані про події до систем SIEM для докладного аналізу та виявлення загроз безпеки.
- ❖ **Автоматизовані заходи відновлення.** Налаштуйте автоматизовані заходи для відновлення стану системи без втручання адміністратора.
- ❖ **Інтеграція з іншими системами управління (ITSM.)** Виводьте автоматично створені та закриті тікети у систему управління обслуговуванням IT для документування та аналізу подій.
- ❖ **Моніторинг результатів автоматизації.** Визначайте показники ефективності та моніторте результати автоматизованих дій.
- ❖ **Вдосконалення сценаріїв.** Постійно вдосконалюйте та адаптуйте автоматизовані сценарії на основі вивчених уроків.

Використовуючи скрипти, автоматизаційні засоби та інтеграцію з іншими системами, можна створити гнучкі та ефективні процеси автоматичної реакції на різноманітні сценарії.

Висновки до розділу

Ми одним оком подивилися на кілька інструментів системного моніторингу з дуже широкого існуючого спектру, який вимагає уважного вибору залежно від конкретних завдань та бюджетних обмежень.

Nagios: Підходить для ефективного моніторингу серверів та мережі, але може вимагати додаткових ресурсів для розширення функціональності.

Zabbix: Відмінно підходить для моніторингу ресурсів та додатків, з високою гнучкістю та можливістю інтеграції.

Prometheus: Ефективний у контейнеризованих середовищах, забезпечує моніторинг та аналіз метрик.

Особливу роль при виборі інструментів системного моніторингу відіграють бюджетні обмеження. Іншими словами: вибір між безкоштовними та комерційними інструментами залежить від фінансових можливостей та потреб користувача.

Навіть дуже поширених універсальних інструментів системного моніторигу існує кілька десятків. Nagios, Zabbix, Prometheus - лише вершина айсберга.

Перед налаштуванням системи моніторингу першорядним завданням залишається глибоке дослідження ринку інструментів, з урахуванням конкретних вимог проекту та фінансових реалій

Контрольні запитання

1) У яких операційних системах зазвичай інсталюється Nagios Core?
2) Який стек технологій використовується для інсталяції Nagios Core?
3) У яких операційних системах зазвичай інсталюється Zabbix?
4) Які стеки підтримує Zabbix для інсталяції?
5) Які існують безкоштовні методи інсталяції Zabbix?
6) У чому відмінність між Zabbix Agent та Zabbix Agent2?
7) Яка архітектура використовується у Nagios Core?
8) Яка роль центрального сервера у Nagios Core?
9) Як відбувається активація запуску плагінів у Nagios Core?
10) Які обов’язкові елементи входять до архітектури Prometheus?
11) Які методи Service Discovery підтримує Prometheus?
12) Яка система моніторингу зберігає дані у вигляді часових рядів?
13) Який основний конфігураційний файл використовується у Nagios Core?
14) Які системи моніторингу поширюються на умовах GNU GPL?
15) З якими джерелами даних може інтегруватися Grafana?
16) Яку основну функцію виконує Grafana у системах моніторингу?

Розділ 6. Аналіз та візуалізація даних мережного моніторингу.

1. Методи аналізу даних мережного моніторингу та їх застосування для виявлення аномалій та проблем в мережі.

Сучасна мережна інфраструктура зростає в розмірах і складності з кожним днем. Збільшення обсягів даних, різноманіття платформ і сервісів, а також зростання кількості користувачів ставлять перед адміністраторами мережі низку викликів. Одним з ключових інструментів для ефективного управління та забезпечення надійності мережі є мережний моніторинг. Але просто збирати дані не достатньо. Не менш важливо аналізувати ці дані для виявлення аномалій, проблем, а також для прогнозування та усунення можливих ризиків.

У цьому розділі ми розглянемо методи аналізу даних мережного моніторингу та їх застосування для виявлення аномалій та проблем в мережі. Також поговоримо про використання інструментів візуалізації даних для створення графіків, діаграм та звітів, які дозволяють нам краще зрозуміти стан мережі та вчасно реагувати на будь-які зміни [59].

Поглибивши розуміння методів аналізу та візуалізації даних мережного моніторингу, ми зможемо забезпечити більш ефективне управління та надійність мережі, що є критично важливим у сучасному світі цифрових технологій [1].

Типи рішень для моніторингу мережі:

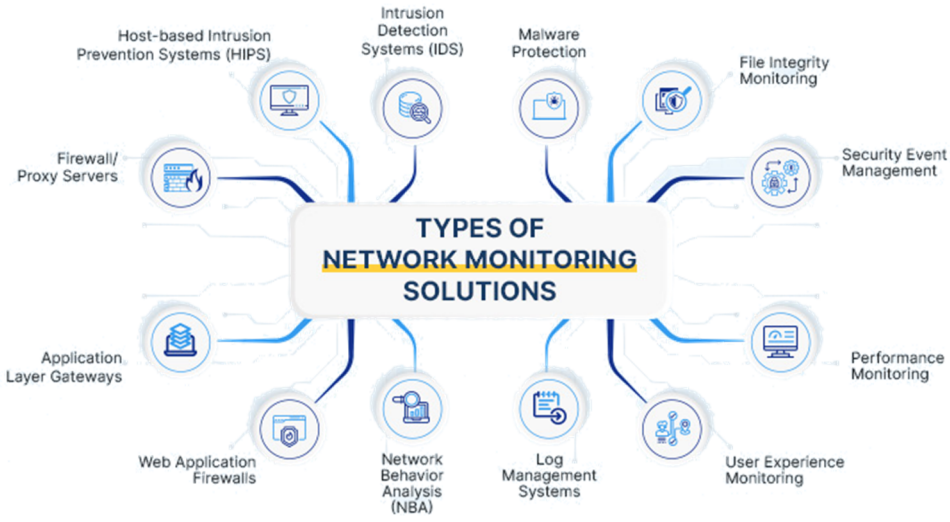


Рис. 6.1. Рішення для моніторингу мережі

На ринку доступно багато видів рішень для моніторингу мережі. Деякі з них наведено на рис. 6.1.

- Системи виявлення вторгнень (IDS)
- Системи управління журналами

- Системи запобігання вторгнень на основі хоста (HIPS)
- Брандмауер/ проксі-сервери
- Шлюзи прикладного рівня
- Брандмауери веб-додатків
- Аналіз поведінки мережі (NBA)
- Моніторинг досвіду користувача
- Контроль продуктивності
- Управління подіями безпеки
- Контроль цілісності файлів
- Захист від шкідливих програм

Рішення для моніторингу мережі дозволяють здійснювати безперервне спостереження за мережною інфраструктурою в режимі реального часу. Швидке виявлення та усунення проблем має вирішальне значення для забезпечення продуктивності та безпеки вашої мережі. Моніторинг допомагає виявити всі проблеми продуктивності, включаючи ті, які не можуть бути зафіксовані стандартними мережними командами. Це дозволяє запобігати збоям, мінімізувати час простою та покращувати ефективність роботи [15].

Основні переваги моніторингу мережі

- **Виявлення кіберзагроз та злочинної діяльності.** Моніторинг мережі дозволяє своєчасно виявляти кіберзагрози, такі як вторгнення, спам, фішингові атаки та інші види злочинної діяльності. Це допомагає запобігти потенційним атакам і захистити мережу від шкідливих дій. Завдяки аналізу трафіку можна ідентифікувати підозрілі активності та запроваджувати відповідні заходи захисту.
- **Підвищення ефективності мережі.** Мережний моніторинг допомагає оптимізувати використання ресурсів, аналізуючи пропускну здатність і навантаження на мережу. Він дозволяє визначити, які пристрої або сервіси споживають найбільше трафіку, та запровадити заходи для покращення продуктивності. Це забезпечує рівномірний розподіл ресурсів і підвищує швидкість роботи мережі.
- **Відновлення роботи мережі після збоїв.** У разі відмови обладнання або програмного забезпечення моніторинг дозволяє швидко ідентифікувати проблему та оперативно її усунути. Це скорочує час простою й допомагає мінімізувати негативний вплив на бізнес-процеси.
- **Покращення безпеки мережі.** Моніторинг допомагає виявляти вразливості в мережній інфраструктурі та вчасно їх усувати, запобігаючи можливим кібератакам. Регулярний аналіз мережного трафіку дозволяє визначати потенційні точки входу для зловмисників і посилювати захист.
- **Підтримка прийняття рішень.** Мережний моніторинг надає цінні аналітичні дані, які допомагають приймати обґрунтовані рішення щодо конфігурації мережі, вибору обладнання та відповідності вимогам

безпеки. Завдяки зібраній інформації можна покращити керування мережною інфраструктурою та оптимізувати її роботу.

- **Виявлення вторгнень у реальному часі.** Коли виникає загроза, моніторинг мережі дозволяє негайно відреагувати та вжити заходів для припинення шкідливої активності. Аналіз мережного трафіку дозволяє виявити джерело атаки та вжити необхідних заходів для захисту від майбутніх інцидентів.
- **Скорочення часу простою, виявлення проблем до їх виникнення.** Простій через збій системи може бути дорогим і призводити до значних фінансових втрат. Використання технології моніторингу мережі допомагає зменшити час простою, адже потенційні проблеми можна виявити та усунути до того, як вони спричинять серйозні збої в роботі [1].
- **Захист організації від кібератак.** Зі зростанням кількості кіберзагроз хакери постійно шукають слабкі місця в мережах компаній. Використовуючи інструменти моніторингу, можна ідентифікувати слабкі місця, що ускладнює несанкціонований доступ для зловмисників.
- **Контроль над мережею.** Моніторинг забезпечує детальну видимість мережного трафіку, дозволяючи контролювати дії користувачів, обмежувати доступ до певних ресурсів і блокувати небажані підключення. Це сприяє кращому керуванню мережею та підвищенню рівня безпеки.
- **Зменшення експлуатаційних витрат.** Моніторинг допомагає виявляти та усувати проблеми ще до того, як вони призведуть до серйозних збоїв або потребуватимуть значних фінансових витрат на їх усунення. Також це сприяє ефективнішому використанню ресурсів, що знижує загальні витрати на підтримку мережі.
- **Запобігання втраті даних через зловмисні атаки.** Один із ключових аспектів безпеки – захист даних від несанкціонованого доступу. Моніторинг дозволяє виявляти спроби хакерських атак, підозрілі дії та витоки інформації. Використання сучасних інструментів моніторингу допомагає підтримувати актуальний рівень безпеки.

У сучасному світі кібербезпека є однією з найважливіших проблем для компаній та окремих користувачів. Кіберзлочинці постійно вдосконалюють свої методи атак, тому захист мережі вимагає проактивного підходу. Моніторинг мережі – це потужний інструмент, що дозволяє виявляти та усувати потенційні загрози ще до того, як вони завдадуть шкоди. Використання технологій моніторингу забезпечує захист від кібератак, допомагає запобігти втраті даних, підвищує продуктивність та оптимізує витрати. Регулярний моніторинг – це не просто додатковий рівень захисту, а необхідна складова стабільної та безпечної мережної інфраструктури.

Чому моніторинг мережі важливий для бізнесу?

Мережний моніторинг **(NM)** має вирішальне значення з кількох причин. Це важливий компонент керування мережею, що дозволяє організаціям оптимізувати продуктивність мережі, мінімізувати час простою та забезпечити безпеку та відповідність своїх мереж.

NM особливо важливий для бізнесу, який займається належним функціонуванням своєї мережі та покладається на хмарні програми, веб-сервіси Інтернету або критичні програми, які здійснюють транзакції через мережу. NM обов'язковий для підприємств із мережами MPLS і SD-WAN і кількома інтернет-ланцюгами та з'єднаннями.

Ось кілька ключових причин, чому моніторинг мережі важливий для бізнесу:

1. **Підвищення продуктивності мережі:** NM є важливим для підтримки високопродуктивної мережі - це головна причина. Це дозволяє виявляти та усувати проблеми для покращення продуктивності мережі .
2. **Завчасне виявлення проблеми з мережею:** проблеми з мережею, як проблеми з доступністю мережі , так і проблеми з продуктивністю, можуть траплятися і трапляються, і вони можуть мати значний вплив на роботу. Тому важливо завчасно виявляти всі проблеми з мережею, перш ніж вони спричинять хаос:
   - ✓ **Проблеми з доступністю:** проблеми з доступністю характеризуються повною втратою з'єднання. Ці типи збоїв у мережі можуть бути спричинені різними факторами, зокрема фізичними пошкодженнями мережної інфраструктури, обривами оптоволокна та перебоями в електропостачанні. Хоча проблеми з доступністю є поганою новиною для надійності мережі , їх найпростіше визначити та усунути.
   - ✓ **Проблеми з продуктивністю:** проблеми з продуктивністю мережі важче виявити та усунути, ніж проблеми з доступністю. Мережа функціонує менш ефективно, ніж очікувалося або потрібно. Це може включати в себе затримки в передачі даних, втрату пакетів, низьку швидкість передачі, або будь-які інші обмеження, які впливають на продуктивність і ефективність мережі.Ці збої можуть бути постійними або періодичними, причому останні є найважчими для виявлення та усунення несправностей. Проблеми з продуктивністю можуть виникнути будь-де, і вони можуть залишатися непоміченими роками.

3. **Оптимізація взаємодії з кінцевим користувачем:** відстежується робота кінцевого користувача під час використання мережних програм і

служб, а також завчасно виявляються та усуваються проблеми з мережею, перш ніж вони розчарують користувачів.

- ✓ Швидше вирішуються проблеми кінцевих користувачів
- ✓ Підвищується рівень задоволеності кінцевих користувачів

4. **Підвищення безпеки мережі:** підвищення безпеки мережі шляхом виявлення потенційних загроз безпеці, таких як незвичайна мережна активність або спроби неавторизованого доступу.
   - ✓ Виявлення потенційних порушеннь безпеки
   - ✓ Зведення до мінімуму ризику втрати даних
   - ✓ Поліпшення дотримання галузевих стандартів і правил

5. **Планування пропускної здатності та оптимізація мережі**. Аналізуючи мережний трафік і використання ресурсів, мережні адміністратори можуть визначити потенційні вузькі місця для:
   - ✓ Планування оновлення потужностей для підтримки майбутнього зростання
   - ✓ Оптимізації продуктивності мережі для задоволення зростаючих вимог
   - ✓ Зменшення витрат на інфраструктуру в майбутньому

6. **Зниження витрат і покращення рентабельності інвестицій:** моніторинг мережі може допомогти знизити витрати та покращити рентабельність інвестицій шляхом виявлення неефективності та оптимізації мережних ресурсів.
   - ✓ Визначення місць з надмірним або недостатнім використанням ресурсів і внесення коректив.
   - ✓ Підвищення рентабельності інвестицій, забезпечивши максимальну продуктивність мережі, що може підвищити продуктивність і задоволеність користувачів.
   - ✓ Зменшення витрат, пов’язаних з простоєм мережі та незапланованими відключеннями.

7. **Підвищення безперервності бізнесу:** нарешті, покращення продуктивності мережі (і взаємодії з користувачем) призведе до загального позитивного впливу на бізнес.
   - ✓ Мінімізується час простою та збої в обслуговуванні, що впливають на продуктивність
   - ✓ Зменшиться кількість збоїв мережі та проблем із продуктивністю, які можуть коштувати часу та грошей підприємствам
   - ✓ Швидко вирішуватимуться проблеми з мережею, не потребуючи додаткових IT-ресурсів

Основні завдання мережного моніторингу

На початку лекції ми обговорили значення мережного моніторингу як з точки зору IT, так і з позиції бізнесу. Важливість цього процесу визначається ключовими завданнями, які він покликаний виконувати. Саме ці завдання формують основні причини необхідності впровадження мережного моніторингу.

- **Аналіз трафіку (Traffic Analysis).** Інструменти мережного моніторингу збирають і аналізують дані про мережний трафік, включаючи обсяг даних, типи трафіку (наприклад, Інтернет, електронна пошта, відео), а також джерела та призначення трафіку. Це допомагає визначити шаблони та аномалії, які можуть вплинути на продуктивність мережі [5].
- **Вимірювання затримки (Latency measurement).** Затримка означає затримку передачі даних через мережу. Відстеження затримки допомагає забезпечити швидку передачу даних без значних затримок, що особливо важливо для додатків у реальному часі, таких як відеоконференції та онлайн-ігри.
- **Втрата пакетів (Packet Loss).** Втрата пакетів відбувається, коли пакети даних скидаються або не досягають місця призначення. Відстеження втрати пакетів допомагає точно визначити проблеми з мережею, які можуть призвести до втрати даних і низької продуктивності програми.
- **Використання пропускної здатності (Bandwidth Utilization).** Інструменти моніторингу відстежують, як використовується пропускна здатність мережі, визначаючи, які програми чи пристрої споживають найбільшу пропускну здатність. Ця інформація є цінною для планування потужності та оптимізації мережі.
- **Справність пристрою (Device Health).** Моніторинг працездатності та стану мережних пристроїв (маршрутизаторів, комутаторів, брандмауерів, серверів тощо) допомагає виявити проблеми з обладнанням або конфігурацією, які можуть вплинути на продуктивність мережі.
- **Безпека (Security).** Моніторинг може відігравати певну роль у безпеці мережі, відстежуючи незвичайну або підозрілу мережну активність, яка може свідчити про порушення безпеки або кібератаку.
- **Попередження та звітування (Alerting and Reporting)**. Інструментарій моніторингу генерує сповіщення, коли перевищуються попередньо визначені порогові значення продуктивності або коли виявляються аномалії. Вони також надають звіти та інформаційні панелі, щоб допомогти мережним адміністраторам і IT-командам візуалізувати й проаналізувати дані про продуктивність мережі [15].
- **Аналіз історичних даних (Historical Data Analysis).** Зберігання історичних даних про продуктивність мережі дає змогу організаціям

визначати тенденції та закономірності з часом, забезпечуючи ефективніше прийняття рішень і проактивне керування мережею.

Ефективний моніторинг продуктивності мережі надає кілька переваг, зокрема підвищення надійності мережі, швидшого вирішення проблем, кращого розподілу ресурсів, покращення взаємодії з користувачем та економії коштів за рахунок ефективного використання мережних ресурсів. Це важлива практика для підтримки загальної працездатності та функціональності сучасних комп'ютерних мереж, особливо на великих підприємствах і в середовищах з інтенсивним об'ємом даних.

У табл. 6.01 наведено опис та застосування загальних методів аналізу даних мережнаого моніторингу.

Таблиця 6.1

Загальні методи аналізу даних мережного моніторингу

Метод	Опис	Застосування
Прослуховування пакетів (Packet sniffing).	Метод полягає в перехопленні і аналізі мережних пакетів, що проходять через мережні пристрої. Інструменти, такі як Wireshark, дозволяють вам переглядати, аналізувати і фільтрувати ці пакети для виявлення аномалій, витрат ресурсів або зловмисних атак.	Перехоплення та аналіз мережних пакетів може допомогти виявити аномальні патерни трафіку, які можуть свідчити про зловживання або атаки на мережу. Наприклад, велика кількість запитів до певного вузла чи сервісу може свідчити про DDoS-атаку.

Трошки більше конкретики про **метод прослуховування пакетів (Packet sniffing)** — це комп'ютерна програма або комп'ютерне обладнання , наприклад пристрій захоплення пакетів, який може аналізувати та реєструвати трафік, що проходить через комп'ютерну мережу або частину мережі. Перехоплення пакетів — це процес перехоплення та реєстрації трафіку. Коли потоки даних проходять мережею, аналізатор фіксує кожен пакет і, якщо необхідно, декодує необроблені дані пакета, показуючи значення різних полів у пакеті, і аналізує його вміст відповідно до відповідних RFC або інших специфікацій.

Аналізатор пакетів, який використовується для перехоплення трафіку в бездротових мережах, називається бездротовим аналізатором

або аналізатором WiFi. Хоча аналізатор пакетів також можна називати мережним аналізатором або аналізатором протоколів, ці терміни також можуть мати інші значення. Технічно аналізатор протоколу може бути ширшим, загальнішим класом, який включає аналізатори/аналізатори пакетів і ми повернемося до цього методу аналізу даних мережного моніторингу пізніше.

У дротових мережах, таких як Ethernet , Token Ring і FDDI , залежно від структури мережі (концентратор або комутатор), може перехопити весь трафік у мережі з однієї машини. У сучасних мережах трафік може бути захоплений за допомогою мережного комутатора за допомогою віддзеркалення портів (Port mirroring), або за допомогою мережного крана (Network tap), що є навіть більш надійним рішенням, ніж використання моніторингового порту для віддзеркалення портів, оскільки мережний кран з меншою ймовірністю скидає пакети під час високого трафіку.

Зустрілось два, можливо незнайомих Вам, терміни:

Віддзеркалення портів (Port mirroring) використовується на мережному комутаторі для надсилання копії мережних пакетів , які бачать один порт комутатора (або весь VLAN), до з'єднання для моніторингу мережі на іншому порту комутатора (Рис. 6.02). Це зазвичай використовується для мережних пристроїв, які потребують моніторингу мережного трафіку, наприклад системи виявлення вторгнень, пасивного зондування або технології моніторингу реального користувача (RUM), яка використовується для підтримки керування продуктивністю додатків (APM). Віддзеркалення портів на комутаторі Cisco Systems зазвичай називають аналізатором комутованих портів (SPAN) або аналізатором віддаленого комутованого порту (RSPAN). Інші постачальники мають різні назви для нього, наприклад, Roving Analysis Port (RAP) на комутаторах 3Com .

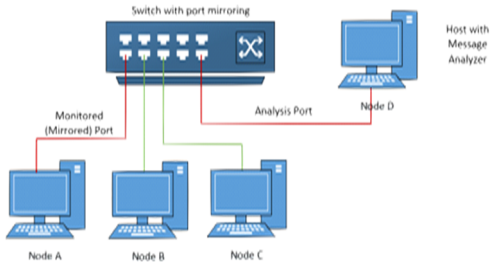


Рис. 6.02. Port mirroring

Мережний кран (Network tap) - це система, яка відстежує події в локальній мережі (Рис. 6.03). Кран зазвичай є спеціальним апаратним пристроєм, який забезпечує спосіб доступу до даних, що

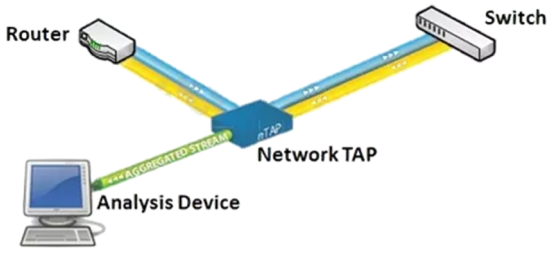


Рис. 6.03. Network tap

передаються через комп'ютерну мережу. Мережний кран має (принаймні) три порти: порт A, порт B і порт монітора. Кран, вставлений між A і B, пропускає весь трафік (вхідний та вихідний потоки даних) безперешкодно в режимі реального часу, але також копіює ті самі дані на порт свого монітора, дозволяючи третій стороні слухати.

Мережні крани у деяких джерелах мають назву мережні відводи, зазвичай використовуються для систем виявлення вторгнень у мережу, запису VoIP, мережного зондування, аналізаторів пакетів та інших пристроїв моніторингу та збору та програмного забезпечення, яким потрібен доступ до сегмента мережі. Мережні крани використовуються в програмах безпеки, тому що вони ненав'язливі, не виявляються в мережі (не мають фізичної чи логічної адреси).

У бездротових локальних мережах трафік можна захоплювати по одному каналу за раз або за допомогою кількох адаптерів по кількох каналах одночасно. Зрозуміло, що 100% трафіку у таких мережах захопити проблематично, хоча теоретично можливо.

У дротових широкомовних і бездротових локальних мережах для захоплення одноадресного трафіку між іншими машинами мережний адаптер , який захоплює трафік, має бути в безладному режимі (promiscuous mode).

У комп'ютерних мережах безладний режим (promiscuous mode) — це режим для контролера інтерфейсу дротової мережі (NIC) або контролера інтерфейсу бездротової мережі (WNIC), який змушує контролер пропускати весь трафік, який він отримує, до центрального процесора (CPU), а не пропускати лише кадри. на отримання яких контролер спеціально запрограмований.

У бездротових локальних мережах, навіть якщо адаптер працює в безладному режимі (promiscuous mode), пакети, які не відповідають набору послуг, для якого налаштовано адаптер, зазвичай ігноруються. Щоб побачити ці пакети, адаптер має бути в режимі моніторингу. Не потрібно ніяких спеціальних налаштувань для захоплення багатоадресного трафіку до групи багатоадресної розсилки, або широкомовного трафіку.

Під час захоплення трафіку записується або весь вміст пакетів, або лише заголовки . Запис лише заголовків зменшує вимоги до пам'яті та дозволяє уникнути деяких юридичних проблем щодо конфіденційності, але часто надає достатньо інформації для діагностики проблем.

Зібрана інформація декодується з необробленої цифрової форми в формат, зрозумілий людині, що дозволяє переглянути інформацію

обміну. Аналізатори протоколів відрізняються за своїми можливостями відображення та аналізу даних.

Деякі аналізатори протоколів також можуть генерувати трафік. Вони можуть виконувати функції тестувальників протоколів. Такі тестери генерують трафік з правильним протоколом для функціонального тестування, а також можуть мати можливість навмисно вводити помилки, щоб перевірити здатність тестованого пристрою обробляти помилки.

Аналізатори протоколів також можуть бути апаратними у форматі зонда, або, що більш поширене, у поєднанні з дисковим масивом. Ці пристрої записують пакети або заголовки пакетів на дисковий масив. Перелічувати їх – нудна справа, згадаю лише всім Вам знайомий Wireshark.

Наступні методи аналізу даних мережного моніторингу це:

Продовження табл. 6.1

Метод	Опис	Застосування
Протокольний аналіз (Protocol Analysis).	Аналіз структури та змісту мережних протоколів для виявлення аномалій або проблем у спілкуванні між пристроями. Протокольний аналіз включає в себе перевірку правильності даних, що обмінюються між пристроями, та виявлення будь-яких відхилень від стандартів.	Шляхом аналізу мережних протоколів можна виявити аномальні зміни в структурі або змісті пакетів, що можуть вказувати на потенційні проблеми в мережі, такі як несправності обладнання або атаки.
Аналіз трафіку (Traffic Analysis).	Аналіз трафіку передбачає вивчення та аналіз мережного трафіку для виявлення та діагностики проблем, пов’язаних із продуктивністю мережі, безпекою та відповідністю. Він передбачає моніторинг і аналіз потоку пакетів даних у мережі, обсягу, швидкості та напрямку мережного трафіку для	Моніторинг обсягу, швидкості та напрямку трафіку дозволяє виявляти аномальні збільшення чи зменшення трафіку, що можуть бути наслідком проблем у мережі або зловмисних дій.

	виявлення потенційних вузьких місць, перевантажень та інших проблем із продуктивністю, які можуть вплинути на продуктивність мережі. Аналіз трафіку може включати аналіз таких показників, як використання пропускної здатності, втрату пакетів, затримку та пропускну здатність. Метод дозволяє виявляти зміни в патернах трафіку, незвичайні обсяги або швидкості, а також потенційні точки перенапруження.	
Виявлення зловмисних дій (Intrusion Detection).	Використання спеціалізованих систем виявлення вторгнень (IDS) або систем виявлення вразливостей (Vulnerability Detection Systems) для виявлення незвичайної або потенційно шкідливої активності в мережі.	Системи виявлення вторгнень використовуються для пошуку підозрілих дій або патернів, які можуть свідчити про зловмисні атаки в мережі, такі як намагання несанкціонованого доступу до системи чи ресурсів.

Конкретика про метод виявлення зловмисних дій (Intrusion Detection). це пристрій або програмне забезпечення, яке відстежує мережу чи системи на наявність зловмисної активності чи порушень політики. Інформація про будь-яке вторгнення або порушення зазвичай повідомляється адміністратору або збирається централізовано за допомогою системи керування інформацією та подіями безпеки (SIEM) . Система SIEM поєднує вихідні дані з кількох джерел і використовує методи фільтрації критичних та некритичних подій, щоб відрізнити зловмисну активність від помилкових спрацювань.

Типи **Intrusion Detection** варіюються від окремих комп’ютерів до великих мереж.

IDS можна класифікувати за місцем виявлення (мережа чи хост) або методом виявлення, який використовується (сигнатурний або на основі аномалії).

За місцем виявлення класифікація наступна:

Найпоширенішими класифікаціями є системи виявлення вторгнень у мережу (NIDS) і системи виявлення вторгнень на основі хосту (HIDS). Система, яка відстежує важливі файли операційної системи, є прикладом HIDS, тоді як система, яка аналізує вхідний мережний трафік, є прикладом NIDS.

- **Системи виявлення мережних вторгнень (NIDS)** розміщуються в стратегічній точці або точках у мережі для моніторингу трафіку до та від усіх пристроїв у мережі. NIDS виконує аналіз трафіку, що проходить у всій підмережі , і зіставляє трафік, який передається підмережами, з бібліотекою відомих атак. Після виявлення атаки або виявлення ненормальної поведінки надсилається сповіщення. Прикладом NIDS може бути встановлення його в підмережі, де розташовані брандмауери, щоб побачити, чи хтось намагається зламати брандмауер. В ідеалі можна сканувати весь вхідний і вихідний трафік, однак це може створити вузьке місце, яке погіршить загальну швидкість мережі. OPNET і NetSim є широко використовуваними інструментами для імітації мережних систем виявлення вторгнень. Системи NID також здатні порівнювати сигнатури подібних пакетів, щоб зв'язувати та видаляти виявлені шкідливі пакети, які мають підпис, що відповідає записам у NIDS. Коли ми класифікуємо дизайн NIDS відповідно до властивості інтерактивності системи, існує два типи: он-лайн і офлайн NIDS, які часто називають вбудованим і кран-режимом відповідно. Он-лайн NIDS працює з мережею в режимі реального часу та аналізує пакети Ethernet і застосовує деякі правила, щоб вирішити, чи це атака чи ні. Off-line NIDS працює зі збереженими даними та пропускає їх через аналітичні процеси для визначення атаки.
- **Системи виявлення вторгнень (HIDS)** працюють на окремих хостах або пристроях у мережі. HIDS відстежує вхідні та вихідні пакети лише з пристрою та сповіщатиме користувача або адміністратора, якщо буде виявлено підозрілу активність. Він робить знімок існуючих системних файлів і порівнює його з попереднім знімком. Якщо критичні системні файли були змінені або видалені, сповіщення надсилається адміністратору для дослідження. Приклад використання HIDS можна побачити на критично важливих машинах, на яких не очікується зміна конфігурацій.

За методом виявлення аномалій існує наступна класифікація:

✓ виявлення на основі сигнатур (розпізнавання поганих шаблонів, наприклад шкідливих програм).
✓ виявлення на основі аномалій (виявлення відхилень від моделі «хорошого» трафіку, яка часто покладається на машинне навчання).
✓ виявлення на основі репутації (розпізнавання потенційної загрози за балами репутації) Цей клас фактично є підкласом виявлення на основі аномалій. .

Трошки детальніше про цю класифікацію:

- **На основі підпису**, або друга назва — **IDS на основі сигнатур** — це виявлення атак шляхом пошуку певних шаблонів, наприклад послідовностей байтів у мережному трафіку або відомих зловмисних інструкцій, які використовуються шкідливим програмним забезпеченням. Ця технологія та термінологія походить від антивірусного програмного забезпечення , яке називає ці виявлені шаблони сигнатурами. Хоча IDS на основі сигнатур може легко виявити відомі атаки, важко виявити нові атаки, для яких відсутні шаблони.

- **Системи виявлення вторгнень на основі аномалій** були введені в першу чергу для виявлення невідомих атак, частково через швидкий розвиток шкідливого програмного забезпечення. Основний підхід полягає у використанні машинного навчання для створення моделі надійної діяльності, а потім порівняння нової поведінки з цією моделлю. Оскільки ці моделі можна навчити відповідно до програм і апаратних конфігурацій, метод на основі машинного навчання має кращу узагальнену властивість порівняно з традиційними IDS на основі підпису. Хоча цей підхід дозволяє виявляти раніше невідомі атаки, він дає доволі високий відсоток хибних спрацьовувань: раніше невідомі законні дії можуть бути класифіковані як зловмисні. Більшість існуючих IDS страждають від того, що процес виявлення займає багато часу, що погіршує продуктивність IDS. Ефективний алгоритм вибору ознак робить процес класифікації, що використовується для виявлення, більш надійним.

На майбутнє, як перспективні теми для обрання дипломної роботи – зверніть увагу на клас безкоштовних систем виявлення атак (англ. Intrusion Detection System, IDS) для захисту MS AD та іншої інфраструктури. Найбільш перспективні в цьому плані:

- **Suricata** є відкритою і доступною для безкоштовного використання - ліцензія GPLv2. Основні характеристики Suricata включають:
➢ ***Швидкість та масштабованість.*** Побудована для ефективної обробки великого обсягу мережного трафіку та може використовувати

багатопоточність для оптимізації продуктивності на системах з багатьма ядрами.

- ***Підтримка великої кількості протоколів.*** Розпізнає і аналізує різні мережні протоколи, включаючи TCP, UDP, ICMP, IPv4, IPv6, HTTP, TLS, DNS і багато інших.
- ***Підтримка правил для виявлення загроз.*** Використовує мову правил для виявлення аномального або підозрілого мережного трафіку. Ці правила можуть бути розроблені спільнотою або створені користувачем, враховуючи конкретні потреби мережі.
- **Багатофункціональність і логування.** Забезпечує широкі можливості логування подій, які можуть бути використані для аналізу та виявлення потенційних загроз.
- ***Оптимізація для використання в хмарних середовищах***. Легко інтегрується з хмарними платформами, що робить інтсурмент практичним для застосування в різноманітних хмарних і гібридних середовищах.

- **Splunk** це платформа для аналізу та моніторингу даних у реальному часі. Використовує ліцензію, яка базується на обсязі оброблених даних (дані, які імпортуються та оброблюються Splunk). Це може бути обмеження обсягу даних на день або на місяць, в залежності від типу ліцензії (Free, Enterprise, Cloud, тощо). Основні характеристики та напрямки використання Splunk включають:
- ***Збір та індексація даних***. Збирає дані з різноманітних джерел, включаючи журнали подій, тексти логів, дані метрик та інші джерела. Після цього він індексує ці дані, щоб забезпечити швидкий пошук та аналіз.
- ***Пошук та аналіз даних***. Має потужний інтерфейс для пошуку, фільтрації та аналізу даних. Користувачі можуть використовувати власні запити та доповнення для вивчення великих обсягів даних.
- ***Візуалізація даних.*** Дозволяє створювати графіки, діаграми та інші візуальні елементи для представлення результатів аналізу даних.
- ***Моніторинг безпеки***. Може використовуватися для моніторингу та виявлення інцидентів безпеки. Здатен агрегувати дані з різних джерел, що дозволяє виявляти аномалії та потенційні загрози.
- ***Моніторинг та аналіз Active Directory***. Використовується для моніторингу та аналізу подій у середовищі Active Directory та може агрегувати дані з лог-файлів AD та інших джерел для виявлення аномалій, атак та інших проблем.
- ***Дозволяє налаштовувати запити та дашборди*** відповідно до конкретних потреб користувача, забезпечуючи гнучкість та масштабованість для різних використань, включаючи моніторинг та захист Active Directory.

- **ELK Stack (Elasticsearch, Logstash, Kibana)** це набір відкритих інструментів для агрегації, аналізу та візуалізації лог-файлів та інших даних. Функціонал та напрямки використання ELK Stack:

➢ ***Elasticsearch функціонал:***

✓ ***Пошук та індексація:*** використовує механізм індексації, щоб швидко забезпечити пошук та аналіз даних.

✓ ***Розподілена архітектура:*** забезпечує розподілену обробку та зберігання даних для масштабованості.

✓ ***RESTful API:*** надає простий RESTful API для запитів до даних та використання власних додатків.

➢ ***Elasticsearch напрямок використання:***

✓ ***Пошук та аналіз лог-файлів***: використовується для швидкого та ефективного пошуку лог-файлів та інших даних.

✓ ***Моніторинг та аналіз продуктивност***і: використовується для моніторингу систем, мереж та інших аспектів продуктивності.

➢ ***Logstash функціонал:***

✓ Потоковий обробник даних: Logstash призначений для обробки, трансформації та вивантаження даних.

✓ Підтримка різних входів та виходів: Підтримує велику кількість вхідних джерел даних та можливостей виведення.

➢ ***Напрямок використання Logstash:***

✓ ***Збір та нормалізація лог-файлів***: Logstash дозволяє збирати дані з різних джерел та нормалізувати їх для подальшого аналізу.

✓ ***Передавання даних в Elasticsearch***: Logstash інтегрується з Elasticsearch, щоб ефективно пересилати оброблені дані для індексації.

➢ ***Kibana. Функціонал:***

✓ Візуалізація даних: Kibana дозволяє створювати графіки, діаграми та інші візуальні елементи на основі даних в Elasticsearch.

✓ Дашборди та звіти: Надає можливості створення і керування дашбордами для моніторингу та аналізу даних.

➢ ***Напрямок використання Kibana:***

✓ ***Моніторинг та аналіз лог-файлів***: Kibana є потужним інструментом для вивчення лог-даних та створення звітів.

✓ ***Візуалізація даних безпеки***: Використовується для відображення статистики та графіків, пов'язаних з безпекою, такими як аналіз входів/виходів, аномалії та інше.

➢ ***IDS в MS AD з ELK Stack:***

✓ ELK Stack може ефективно використовуватися для виявлення інтрузій у середовищі Active Directory. Logstash може збирати та нормалізувати журнали подій AD, а потім передавати їх до Elasticsearch для індексації. Kibana дозволяє створювати дашборди та візуалізації для моніторингу безпеки, виявлення аномалій та аналізу подій. ELK Stack

може бути налаштований для інтеграції з різними іншими інструментами безпеки та IDS, що робить його ефективним для цілей безпеки в AD.

- **Snort** – система виявлення і запобігання інтрузійним діям (Intrusion Detection and Prevention System, IDS/IPS), що використовується для моніторингу та аналізу мережного трафіку з метою виявлення потенційних загроз безпеці та розповсюджується під GNU General Public License (GPL)

➢ Snort може бути успішно використаний для виявлення атак та моніторингу безпеки в середовищі Active Directory (AD). Він аналізує мережний трафік і виявляє потенційно шкідливі або небезпечні дії.

➢ Щоб налаштувати Snort для роботи з AD, вам слід визначити специфічні правила для виявлення аномалій та загроз, що можуть виникнути в середовищі AD, хоча існують і набори шаблонів визначення правил.

➢ Використання Snort у поєднанні з іншими інструментами аналізу та збору лог-файлів може забезпечити комплексний підхід до моніторингу та захисту AD.

➢ Популярний та ефективний інструмент для IDS/IPS та моніторингу безпеки Відкрита ліцензія робить Snort доступним для різних типів організацій, включаючи ті, які використовують середовище Active Directory.

Продовження табл. 6.1

Метод	Опис	Застосування
Побудова та аналіз графіків (Graph Analysis).	Використання графіків для візуалізації зв'язків між різними мережними вузлами та їх активністю. Це може допомогти виявити аномальні мережні зв'язки або патерни спілкування.	Дозволяє візуалізувати зв'язки між мережними вузлами та активністю. Наприклад, зміни в графіку можуть вказувати на аномальні зв'язки між вузлами, що може свідчити про атаки або несправності.

На Рис. 6.04 представлений зразок мережного графіка з Вікіпедії, що відображає внески редакторів у різні мовні розділи енциклопедії. Використовуючи цей приклад, згадаємо основні концепції теорії графів (або коротко нагадаємо їх, якщо ви вже знайомі з цією темою):

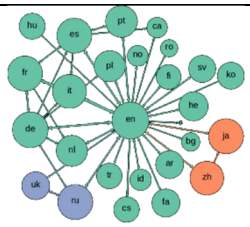


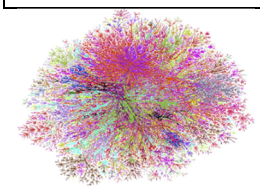
Рис. 6.4. Wiki map

- ✓ Вершини (вузли) — це кола, що представляють різні мови, якими були написані статті.
- ✓ Ребра — лінії, що з'єднують вершини. Кожне ребро визначається через функцію інцидентності, яка встановлює зв'язок між парою вершин.

Такий підхід до візуалізації зв'язків допомагає аналізувати взаємодію між мережними вузлами та їх активність.

Продовження табл. 6.1

Метод	Опис	Застосування
Моделювання мережі (Network Modeling).	Використання математичних моделей для аналізу різних аспектів мережі, таких як її структура, пропускна здатність та вразливості. Це дозволяє прогнозувати поведінку мережі та виявляти можливі проблеми заздалегідь.	Дозволяє створити математичні моделі її структури, пропускної здатності та поведінки. Ці моделі можуть використовуватися для симуляції різних сценаріїв та виявлення можливих проблем або вразливостей. Наприклад, можна моделювати вплив відмови певних вузлів чи ліній на загальну пропускну здатність мережі або виявити точки перенапруження.



Цей метод проілюстровано інтернет-картою 2004 року. Кольорова хмарка, це математична модель мережі з'єднань між пристроями в Інтернеті. Кольори позначають оператора мережі. Структура визначається надсиланням шторму IP-пакетів випадковим чином через мережу.

Продовження табл. 6.1

Метод	Опис	Застосування
Статистичний аналіз (Statistical Analysis).	Використання статистичних методів для виявлення патернів, аномалій або важливих змін в мережних даних. Це може включати аналіз розподілу даних,	Застосування статистичних методів дозволяє виявляти аномалії у мережних даних на основі їх розподілу, кореляцій та інших параметрів.

	кореляційні аналізи та інші методи.	

Гарна ілюстрація до методу статистичного аналізу є звіт використання інтернет трафіку, створений за допомогою SolarWinds ® NetFlow Traffic Analyzer (NTA). Тут представлена статистична вибірка Топ-5 застосунків і топ-10 одержувачів (Рис. 6.05).

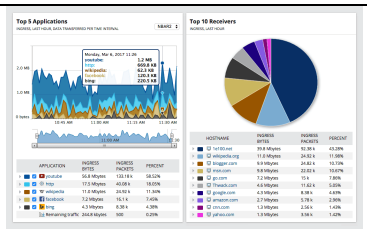


Рис. 6.05. Топ-5 застосунків та топ-10 одержувачів.

SolarWinds® NetFlow Traffic Analyzer (NTA) — це потужний інструмент для моніторингу мережного трафіку, який входить до складу SolarWinds Network Performance Monitor (NPM). Основне завдання NTA — аналіз потокових даних (NetFlow, sFlow, J-Flow, IPFIX та ін.) для виявлення аномалій, визначення основних споживачів трафіку та оптимізації роботи мережі.

SolarWinds NetFlow Traffic Analyzer (NTA) є комерційним продуктом. Він поставляється як доповнення до SolarWinds Network Performance Monitor (NPM), тобто без NPM його використовувати не можна.

Продовження табл. 6.1

Метод	**Опис**	**Застосування**
Історичний аналіз даних про продуктивність мережі (Historical analysis of NPD).	Аналіз історичних даних є ключовим компонентом у вирішенні минулих проблем із продуктивністю мережі, особливо періодичних. Він передбачає збір і аналіз історичних даних продуктивності за певний період часу для виявлення тенденцій і закономірностей, пов'язаних з продуктивністю мережі.	Дані історичного аналізу можуть бути використані для виявлення закономірностей і тенденцій, пов'язаних з продуктивністю мережі, таких як зміни в моделях трафіку, періоди пікового використання та повторювані або періодичні проблеми. Це також дозволяє IT-командам вирішувати проблеми, які виникли в минулому, а також передбачати майбутні проблеми з продуктивністю мережі та планувати оновлення потужності або інші покращення мережі.
Відображення топології мережі	Відображення топології мережі	Після створення карти топології мережі IT-

(Network topology display)	забезпечує створення та підтримку карти або діаграми, яка представляє фізичне та логічне розташування мережі. Відображення топології мережі дає можливість IT-командам візуалізувати та розуміти зв'язки між мережними засобами, програмами та користувачами, що може допомогти IT-командам виявити поточні проблеми з продуктивною мережею, загрози безпеки та ризики відповідальності.	спеціалісти можуть використовувати її для виявлення продуктивності наявних вузьких місць, таких як перевантажені комутатори або маршрутизатори, і для оптимізації продуктивності мережі.
Моніторинг продуктивності додатків (Application Performance Monitoring) (APM)	Моніторинг продуктивності додатків (APM) зосереджується на моніторингу та аналізі продуктивності конкретних програм, що працюють у мережі. Він передбачає моніторинг і аналіз показників продуктивності додатків, таких як час відгук, затримка, пропускна здатність і частота помилок, щоб виявити	APM передбачає збір даних із різних джерел, включаючи журнали додатків, журнали серверів, мережний трафік і взаємодію користувачів, щоб забезпечити цілісне уявлення про продуктивність додатків.

	проблеми, пов’язані з продуктивністю додатків, доступністю та взаємодією з користувачем.	

Традиційний моніторинг додатків можна розділити на два різні типи для всіх намірів і цілей: APM, зосереджений на загальному стані програми, і розширений APM, який забезпечує широкі показники працездатності програми аж до рівня коду. Хоча промисловість офіційно не розрізняє ці підходи, корисно знати різницю.

1. **APM, орієнтований на загальний стан додатків**. Більшість програм сьогодні пропонують різні показники продуктивності, як-от Microsoft, яка пропонує розширений інтерфейс API для кожного продукту. API надають дані про працездатність додатків, що дозволяє вимірювати продуктивність додатків. Оскільки метрики доступні через rest API, можна запитувати ці дані так само, як ви запитували б стандартні мережні дані, такі як SNMP . Коли програма надає дані, ви можете розмістити їх на інформаційних панелях, діаграмах, віджетах, щоб розповісти історію про її ефективність. Як правило, постачальники APM, пропонуватимуть шаблони додатків із інформаційними панелями, дозволяючи користувачам встановлювати шаблони з бібліотеки, вибирати, що контролювати, і просто чекати, поки інструмент збере дані з API для обраного продукту.
2. **Розширений APM.** Цей тип моніторингу продуктивності застосунку використовує розширений, вичерпний і складніший підхід до визначення працездатності застосунку аж до компонента застосунку та коду, з якого він складається. Найчастіше користувачами передових рішень APM є компанії, що займаються розробкою програмного забезпечення, яким необхідно постійно стежити за справністю свого коду.

Продовження табл. 6.1

Метод	Опис	Застосування
Моніторинг мережних пристроїв SNMP (Simple Network Management Protocol) SNMP.	SNMP дозволяє IT-командам контролювати продуктивність і стан мережних пристроїв/обладнання.	Пристрої з увімкненням SNMP можна налаштувати для зв’язку з сервером SNMP, який відповідатиме за збір цінних даних про працездатність і

		використання ресурсів мережного пристрою.
Машинне навчання та штучний інтелект (Machine Learning and Artificial Intelligence).	Використання алгоритмів машинного навчання та штучного інтелекту для автоматизованого аналізу мережних даних. Це може включати виявлення аномалій, класифікацію трафіку та передбачення подій в мережі.	Використання алгоритмів машинного навчання для автоматизованого виявлення аномалій у мережі на основі історичних даних та патернів поведінки.

2. Типи інструментів моніторингу мережі

Для моніторингу мережі використовуються різноманітні інструменти для моніторингу та аналізу мережного трафіку, виявлення вузьких місць і усунення проблем.

Одним із найважливіших аспектів моніторингу продуктивності мережі є використання спеціалізованих інструментів, призначених для збору та аналізу мережних даних у режимі реального часу, оскільки ви не можете стежити за всім самостійно. Ці інструменти мають багато форм і використовуються мережними адміністраторами, інженерами та IT-фахівцями для моніторингу стану мережі , усунення проблем і оптимізації продуктивності.

З різних типів інструментів моніторингу мережі , які зазвичай використовуються в сучасних мережах виділяють кілька класів інструментів. Розглянемо їх детальніше.

- **Інструменти пасивного моніторингу мережі**

Інструменти пасивного моніторингу мережі збирають і аналізують дані про мережний трафік під час його проходження через мережу.

На відміну від інструментів активного моніторингу, які генерують синтетичний трафік, інструменти пасивного моніторингу фіксують і аналізують трафік, який уже проходить через мережу.

Інструменти пасивного моніторингу мережі зазвичай використовуються для захоплення мережного трафіку, вимірювання показників продуктивності мережі та забезпечення видимості моделей використання мережі .

Інструменти пасивного моніторингу мережі зазвичай включають:

- ✓ **Збір і аналіз реального трафіку** користувачів.
- ✓ **Аналіз заголовків пакетів**.
- ✓ **Аналіз корисного навантаження** для вимірювання показників продуктивності мережі, таких як затримка, тремтіння та пропускна здатність.
- ✓ Забезпечення детальної **видимості шаблонів мережного трафіку**, які можна використовувати для визначення тенденцій, усунення проблем і моніторингу продуктивності додатків.
- ✓ **Виявлення програм, які потребують великої пропускної здатності.**
- ✓ **Аналіз моделей трафіку**
- ✓ **Визначення точок перевантаження** мережі або загального перевантаження мережі .

Переваги пасивного моніторингу

- ✓ Не створює додаткового навантаження на мережні пристрої.
- ✓ Простий у налаштуванні та підтримці.
- ✓ Підходить для моніторингу великих мереж.

Інструменти пасивного моніторингу продуктивності мережі мають кілька недоліків, тому їх все частіше замінюють рішеннями для активного моніторингу. **Ось деякі з недоліків**:

- ✓ **Нездатність генерувати трафік**. Вони можуть бути не в змозі визначити проблеми, які присутні лише в періоди інтенсивного трафіку.
- ✓ **Обмежений обсяг аналізу**. Оскільки вони можуть аналізувати лише зафіксований трафік, це може не надати повної картини продуктивності мережі.
- ✓ **Ресурсовитратні**. Вимагають значних ресурсів для захоплення та зберігання мережного трафіку, що може призвести до проблем із продуктивністю в контрольованій мережі.
- ✓ **Потенційні проблеми з конфіденційністю**. Оскільки вони захоплюють увесь мережний трафік, включаючи конфіденційну інформацію, таку як паролі та конфіденційні дані. Це може викликати проблеми з конфіденційністю, якщо отримані дані не захищені належним чином.
- ✓ **Складність**. Складно використовувати, бо для інтерпретації зібраних даних потрібен високий рівень технічної експертизи.

- ➢ **<u>Інструменти активного моніторингу мережі</u>**

Інструменти активного моніторингу мережі відстежують продуктивність мережі, надсилаючи пакети даних через мережу для імітації трафіку користувача та тестування продуктивності мережі.

Ці інструменти використовуються для активного моніторингу та вимірювання продуктивності мережі, виявлення проблем у режимі реального часу та створення сповіщень, коли продуктивність мережі падає нижче прийнятного рівня.

Інструменти активного моніторингу мережі зазвичай включають:

- ✓ **Розгортання агентів моніторингу** або зондів у ключових місцях мережі для наскрізного моніторингу.
- ✓ **Агенти або зонди безперервно надсилають синтетичні пакети даних через мережу** для вимірювання затримки мережі, пропускної здатності та втрати пакетів.
- ✓ **Дозволяють** мережним адміністраторам **імітувати поведінку користувачів** і перевіряти, як мережа реагує за різних навантажень і умов.

Інструменти активного моніторингу мережі також все частіше замінюють старі рішення пасивного моніторингу з різних причин, таких як:

- ✓ **Проактивний моніторинг**. Імітують трафік користувачів і постійно відстежують мережу на предмет потенційних проблем, що дозволяє IT-командам виявляти та вирішувати проблеми до того, як вони вплинуть на кінцевих користувачів.
- ✓ **Немає захоплення пакетів**. Інструменти активного моніторингу не фіксують трафік реального користувача, що означає, що немає проблем із конфіденційністю даних реального користувача. Це також означає, що вони можуть контролювати продуктивність навіть у періоди низького трафіку.
- ✓ **Точні вимірювання.** Активні інструменти можуть забезпечити точні вимірювання показників продуктивності мережі, таких як затримка, використання пропускної здатності та втрати пакетів, оскільки вони генерують власний трафік і вимірюють відповідь.
- ✓ **Повна видимість.** Можуть контролювати всі компоненти мережі, включаючи сервери, комутатори, маршрутизатори та програми, надаючи IT-командам повну видимість продуктивності всієї мережі.
- ✓ **Гнучкість.** Можна налаштувати відповідно до конкретних потреб моніторингу та розгортати в різних місцях, включаючи хмару, локальні та віддалені місця.
- ✓ **Масштабованість.** Можуть масштабуватися для моніторингу великих і складних мереж із кількома розташуваннями, що робить їх ідеальними для організацій корпоративного рівня.

Недоліки активного моніторингу:

- ✓ Може створювати додаткове навантаження на мережні пристрої.
- ✓ Може бути складним налаштувати та підтримувати.
- ✓ Може не підходити для моніторингу великих мереж.

Найбільш поширені приклади активних інструментів моніторингу мережі Вам знайомі. Це Ping, Traceroute. Вони є вбудованими інструментами, доступними в більшості операційних систем та систем моніторингу. Існують більш просунуті комерційні інструменти, що відстежують продуктивність мережі за допомогою агентів моніторингу, які надсилають синтетичний TCP/UDP-трафік з певною періодичністю між хостами-агентами через ключові місця мережі

- **Засоби моніторингу мережі на основі SNMP**

Інструменти моніторингу мережі на основі SNMP використовують простий протокол керування мережею (SNMP) для моніторингу та керування мережними пристроями .

SNMP — протокол, який використовується для керування мережними пристроями та дозволяє відстежувати та контролювати мережні пристрої. Інструменти моніторингу мережі на основі SNMP можуть збирати інформацію про мережні пристрої, як-от використання ЦП, пам'яті, статистику трафіку та інші параметри.

Інструменти моніторингу на основі SNMP можуть бути як активними, так і пасивними, і їх можна використовувати для моніторингу дротових і бездротових мереж.

- **Інструменти моніторингу продуктивності додатків (APM).**

Інструменти моніторингу продуктивності додатків - Application Performance Monitoring (APM) — це програми, призначені для моніторингу продуктивності та доступності програм у режимі реального часу.

Інструменти APM використовуються для виявлення, діагностики та вирішення проблем продуктивності, які впливають на продуктивність програми та взаємодію з користувачем.

Інструменти APM забезпечують видимість усього стеку додатків, включаючи код додатків, веб-сервери, бази даних та інші компоненти інфраструктури, збираючи різноманітні показники, пов'язані з продуктивністю додатків, наприклад час відповіді, пропускну спроможність, частоту помилок і використання ресурсів, щоб надати розуміння поведінки додатків і оптимізації продуктивності. Вони також мають такі функції, як зіставлення додатків, відстеження транзакцій і діагностика на рівні коду, щоб допомогти виявити та діагностувати проблеми продуктивності.

- **Інструменти моніторингу роботи кінцевого користувача**

Інструменти моніторингу взаємодії з кінцевим користувачем - End-User Experience Monitoring Tools (EUEM) призначені для відстеження досвіду роботи кінцевих користувачів під час використання ними програми або доступу до веб-сайту .

Вони забезпечують видимість того, як програми та служби працюють для кінцевих користувачів, незалежно від того, звідки вони до них звертаються, і можуть допомогти виявити та усунути проблеми, пов'язані з продуктивністю мережі.

Інструменти EUEM збирають дані про різноманітні показники взаємодії з користувачем, включаючи час завантаження сторінки, час відповіді та частоту помилок, серед іншого. Деякі інструменти EUEM також пропонують можливість імітувати взаємодію користувача з додатками, що дозволяє IT-командам виявляти потенційні проблеми, перш ніж вони вплинуть на кінцевих користувачів.

- **Синтетичні засоби моніторингу продуктивності мережі**

Інструменти синтетичного моніторингу продуктивності мережі призначені для імітації поведінки користувача в мережі для вимірювання продуктивності мережі. Ці інструменти генерують синтетичний трафік, який надсилається через мережу, і вимірюють різні показники продуктивності, такі як затримка, втрата пакетів і тремтіння.

Інструменти синтетичного моніторингу використовуються для тестування продуктивності нової мережної інфраструктури, а також виявлення та діагностики проблем у існуючій мережній інфраструктурі. Вони також можуть бути використані для моделювання реальних сценаріїв, таких як велике навантаження користувачів або певні типи мережного трафіку, щоб перевірити продуктивність мережі за різних умов.

Інструменти активного моніторингу продуктивності мережі часто також є синтетичними інструментами моніторингу.

Загалом синтетичні інструменти моніторингу продуктивності мережі забезпечують спосіб проактивного моніторингу та тестування продуктивності мережі для забезпечення оптимальної роботи мережі.

- **Інструменти аналізатора мережних пакетів**

Аналізатори мережних пакетів, також відомі як аналізатори пакетів, аналізатори протоколів або мережні аналізатори, збирають і аналізують пакети даних, надіслані та отримані через мережу.

Інструмент дозволяє мережним адміністраторам і фахівцям із безпеки перевіряти мережний трафік, визначати джерело й призначення пакетів, а також аналізувати вміст пакетів, щоб усунути проблеми з мережею, оптимізувати продуктивність мережі, а також виявити й запобігти загрозам безпеки.

Аналізатори пакетів можна використовувати в дротових і бездротових мережах, і вони підтримують різноманітні мережні протоколи та інтерфейси, такі як Ethernet, Wi-Fi, TCP/IP і HTTP.

- **Інструменти моніторингу мережі на основі потоку**

Інструменти моніторингу мережі на основі потоку збирають інформацію про мережний трафік, аналізуючи дані потоку. Дані потоку — це метадані, які описують характеристики мережного трафіку, наприклад IP-адреси джерела та призначення, номери портів і обсяг переданих даних.

Основи використання протоколів NetFlow та sFlow у мережному моніторингу було описано у одному з попередніх розділів. Моніторинг на основі потоку використовується для визначення типів трафіку, який проходить мережею, а також кінцевих точок, які генерують і отримують трафік.

- **Хмарні інструменти моніторингу мережі**

Хмарні інструменти моніторингу мережі дозволяють контролювати продуктивність мережі з віддаленого місця . Вони розміщені в хмарі та використовують мережу серверів для збору та аналізу даних про мережний трафік, продуктивність програм та інші відповідні показники.

Доступ до хмарних інструментів моніторингу мережі можна отримати через веб-інтерфейс, що полегшує IT-фахівцям моніторинг продуктивності своєї мережі з будь-якого місця, де є підключення до Інтернету. Вони також часто засновані на підписці та можуть пропонувати такі функції, як моніторинг у реальному часі, сповіщення та звітування. Вони також можуть легко масштабуватися для мереж, що ростуть, запропонувати високу доступність і надійність і допомогти організаціям зменшити витрати на IT-інфраструктуру, усуваючи

необхідність купувати та обслуговувати обладнання та програмне забезпечення на місці.

- **Інструменти моніторингу продуктивності Wi-Fi**

Інструменти моніторингу продуктивності Wi-Fi відстежують і аналізують продуктивність мереж Wi-Fi . Вони збирають і аналізують дані з пристроїв Wi-Fi і точок доступу, надаючи інформацію про різні показники продуктивності Wi-Fi, такі як потужність сигналу, пропускна здатність, затримка та втрата пакетів.

Інструменти моніторингу продуктивності Wi-Fi допомагають виявити та усунути проблеми з продуктивністю Wi-Fi, оптимізувати роботу мережі Wi-Fi та забезпечити високу якість надання послуг Wi-Fi кінцевим користувачам.

3. Використання інструментів візуалізації даних для створення графіків, діаграм та звітів.

Графічне відображення інформації допомагає донести потрібну думку, підкріпити сформульований висновок або підкреслити акцент

Одна з труднощів, яка істотно сповільнює та ускладнює сприйняття звітів та аналітичну роботу, полягає в підборі правильного типу діаграми. Невірний її вибір може викликати плутанину в голові у користувача, або системного адміністратора, чи призвести до помилкової інтерпретації даних [19].

Щоб створити діаграму, яка пояснює та демонструє точну аналітику, спочатку потрібно зрозуміти причини, через які взагалі вона може знадобитися. Розглянемо п'ять типових питань, що виникають при виборі типу діаграми. Потім ми дамо огляд основних видів діаграм, що використовуються для відображення результатів моніторингу.

- **5 питань, які потрібно задати собі при виборі діаграми** [19]

1. ***Вам потрібно порівнювати величини?***
   Графіки ідеально підходять для порівняння одного або декількох наборів величин, і вони можуть легко відображати самі низькі і високі показники. Для створення порівняльної діаграми використовуйте наступні типи: гістограма, кругова діаграма, точкова діаграма, шкала зі значеннями.
2. ***Ви хочете показати структуру чогось?***
   Наприклад, ви хочете отримати поточний стан CPUз прив’язкою до типів хостів, або загальний обсяг інтернет трафіку, розбитий на лінки від різних провайдерів. Щоб показати структуру, використовуйте

наступні діаграми: кругова діаграма, гістограма з накопиченням, вертикальний стек, обласна діаграма, діаграма-водоспад.

3. ***Ви хочете зрозуміти, як розподіляються дані?***
   Таблиці з розподілом допомагають зрозуміти основні тенденції і відзначити, що виходить за рамки. Для відображення розподілу даних використовуйте ці діаграми: точкова діаграма, лінійна діаграма, гістограма.

4. ***Ви зацікавлені в аналізі тенденцій у певному наборі даних?***
   Якщо ви хочете дізнатися більше про те, як дані поводяться протягом конкретного часового періоду, є типи діаграм, які дуже добре це відображають. Вам знадобляться: лінійна діаграма, подвійна вісь (стовпець і лінія), гістограма.
5. ***Хочете краще зрозуміти взаємозв'язок між встановленими значеннями?***
   Взаємопов'язані графіки підходять для того, щоб показати, як одна змінна відноситься до іншої або декількох різних змінних. Це можна використовувати, щоб показати позитивний, негативний або нульовий вплив на іншу цифру. Використовуйте для цього такі діаграми: точкова діаграма, бульбашкова діаграма, лінійна діаграма.

Розглянемо типи графіків та діаграм, які використовуються для візуалізації даних у мережному моніторингу. Візуалізація даних є надзвичайно важливим етапом у розумінні стану мережі, виявленні аномалій, аналізі та прийнятті рішень. Ось деякі типи графіків та діаграм, які часто використовуються у мережному моніторингу [1]:

- **Часові графіки**: Ці графіки показують зміну параметрів мережі в залежності від часу (Рис. 6.06). Вони дозволяють виявити тенденції та циклічність у поведінці мережі, а також виявляти винятки та аномалії, які можуть виникнути протягом певного часового періоду.
  Моніторинг у режимі реального часу є критично важливим аспектом моніторингу мережі (NM), який стосується постійного моніторингу показників продуктивності мережі, таких як використання пропускної здатності, затримка, втрата пакетів та інші ключові показники ефективності (KPI) у режимі реального часу. Це дозволяє IT-командам завчасно виявляти проблеми з мережею до того, як вони вплинуть на кінцевих користувачів.

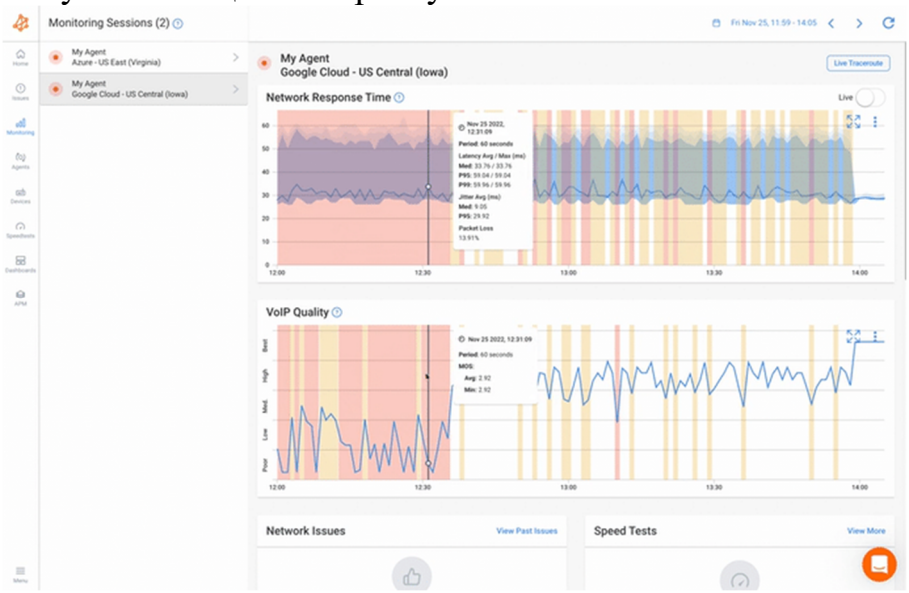


Рис. 6.06. Приклад часового графіку

- **Лінійні графіки** використовуються для відображення залежності між двома змінними, такими як швидкість передачі даних, витрата пропускної здатності тощо (Рис. 6.7). Вони часто використовуються для моніторингу пропускної здатності та швидкості мережі.

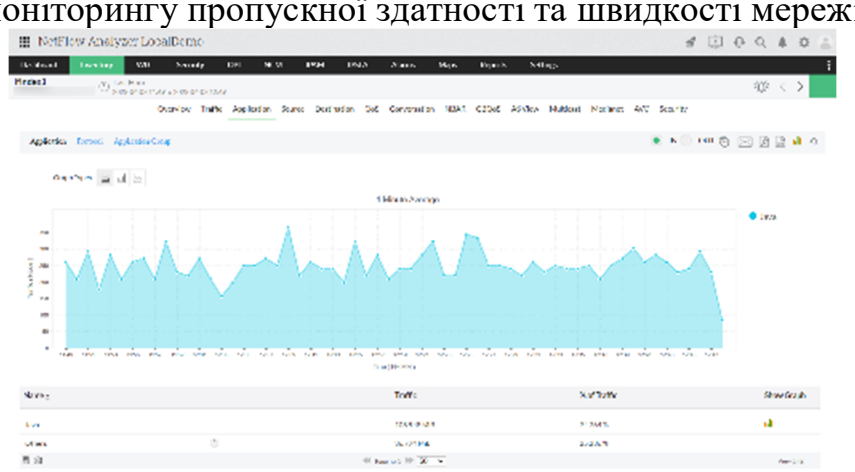


Рис. 6.7. Приклад лінійного графіку

- **Гістограми** – графіки, які показують розподіл даних за певними параметрами. Наприклад, гістограми можуть бути використані для відображення розподілу величини пакетів за їхньою довжиною, що допомагає спостерігати за нормальністю процесу, або виявити незвичайні або аномальні значення (Рис. 6.8).

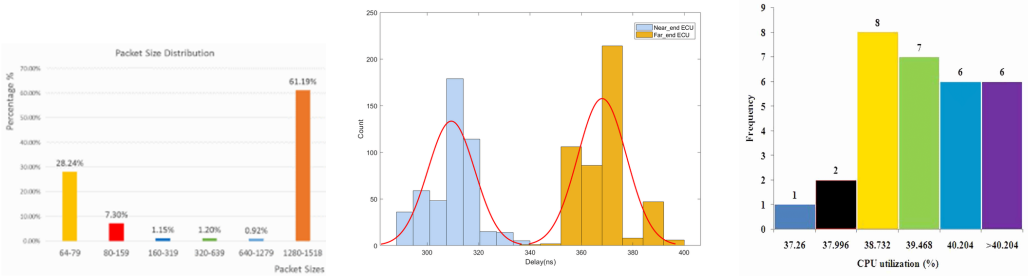


Рис. 6.8. Гістограми розподілу розмірів пакетів, розподілу затримки та використання ЦП сервера

- **Кругові діаграми** часто використовуються для відображення часток різних типів даних у загальному наборі даних (Рис. 6.9). Наприклад, кругова діаграма може показати співвідношення різних протоколів, що використовуються у мережі. Для цього типу діаграм необхідно розуміти, що вони наглядні та прості у розумінні тільки при порівняно невеликій кількості показників, або порівнянні якихось домінуючих показників з залишком кола, що припадає на інші.

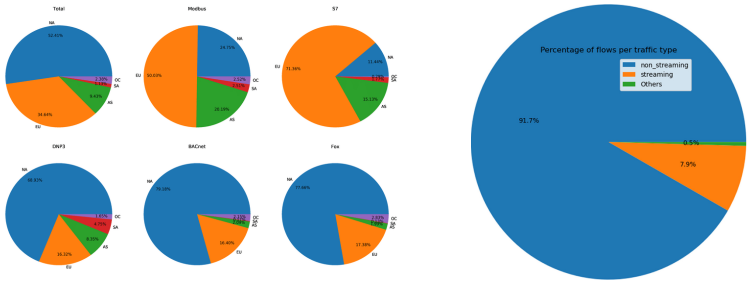


Рис. 6.9. Розподіл мережного трафіку за протоколом та діаграма розподілу трафіку за адресою

- **Теплові карти** використовуються для візуалізації інтенсивності певного параметру на основі розташування у просторі (Рис. 6.10 та 6.11). Наприклад, теплова карта може відображати рівень трафіку на різних ділянках мережі.

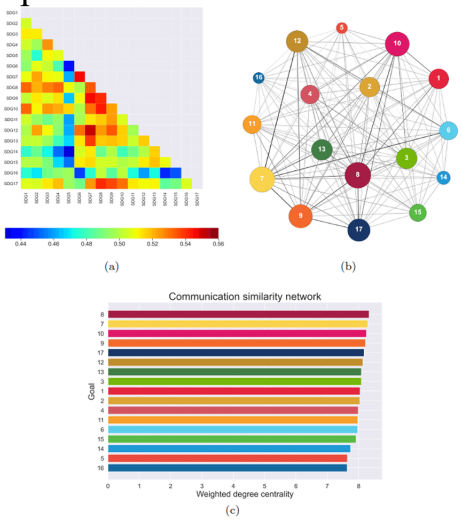


Рис. 6.10. Теплова карта доступності мережі

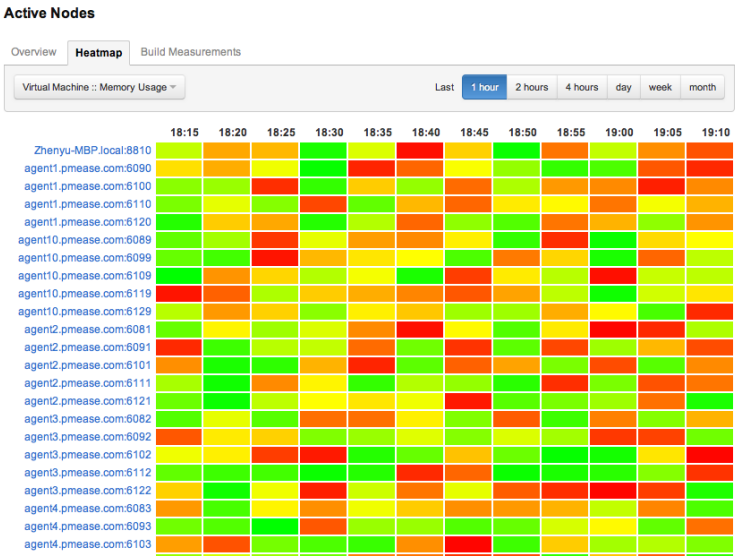


Рис. 6.11. Теплова карта, що показує використання пам'яті кількох серверів

- **Діаграми Ганта** використовуються для відображення розкладу різних подій або завдань у часі (Рис. 6.12). Це може бути корисно для відстеження виконання певних мережних проектів або технічних робіт.

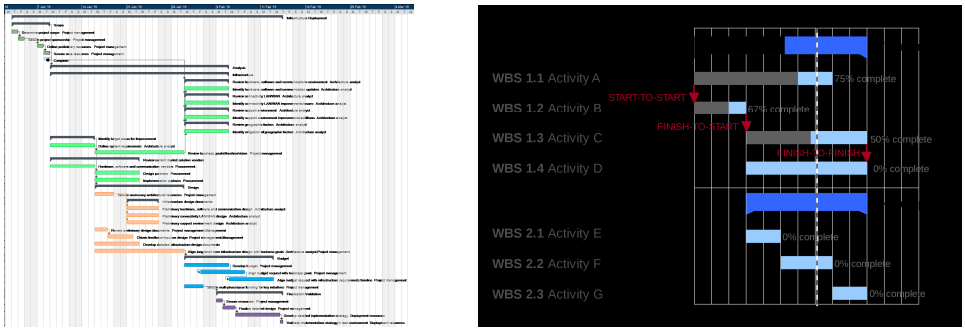


Рис. 6.12. Діаграми розгортання нового маршрутизатора та усунення несправностей

- **Скатерограми** – графіки, що відображають взаємозв'язок між трьома або більше змінними (Рис. 6.13 та 6.14). Вони дозволяють виявити кореляції або залежності між різними параметрами мережі.

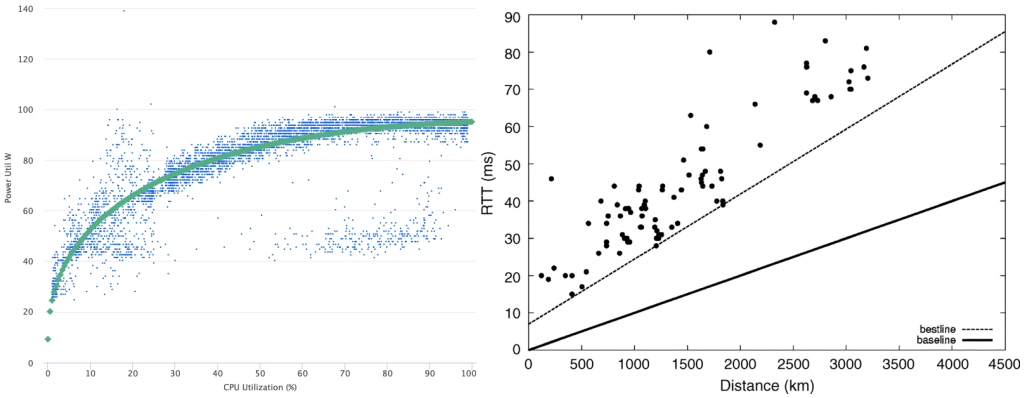


Рис. 6.13. Скатерограми залежності використання ЦП сервера від навантаження та залежності часу відгуку веб-сайту від затримки.

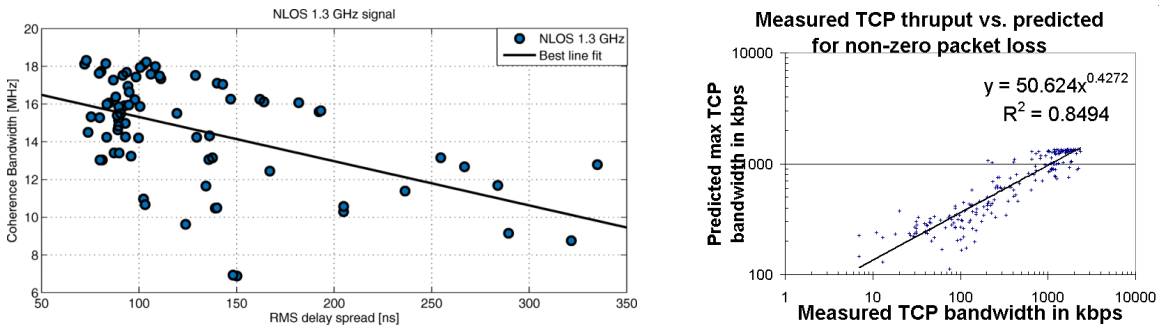


Рис. 6.14. Скатерограми залежності затримки від пропускної здатності та залежності втрати пакетів від пропускної здатності.

Не всі ці типи графіків та діаграм підтримуються кожним інструментом моніторингу, але цей короткий опис має допомогти Вам при обранні типу, якщо він доступний.

4. Розробка звітів та дашбордів для ефективного моніторингу мережі

Термін «dashboard» з англійської на українську перекладається як інформаційна панель. А по суті, дашборд - це програмне рішення, що дозволяє створювати, отримувати та аналізувати дані в реальному часі. Видані інформаційною панеллю «розумні звіти» допомагають власнику, керівнику, менеджеру розуміти певні тенденції в конкретному сегменті діяльності та контролювати події, що відбуваються [59], [15].

Виграш від використання в роботі не класичного звіту, а дашборду в тому, що глобальні і просто важливі показники надаються до вивчення зрозуміло і наочно. Інформаційна панель істотно спрощує сприйняття користувачем складних і різносторонніх відомостей. Той, хто отримує дані за допомогою дашборда, може оцінити поточний стан справ на конкретний момент часу з першого погляду.

Щоб було простіше зрозуміти, що таке дашборд, наведемо кілька прикладів використання подібного програмного рішення. Пригадуєте гігантські монітори з художніх фільмів про глобальні катастрофи або надпотужні випробування? На стіні-дисплеї з'являються зведення, важливі показники про чисельність, фактичне розташування і так далі. Подібні монітори з оперативною інформацією зустрічаються не тільки в кіно. У реальному житті безліч прикладів використання дашборду для своєчасного отримання та аналізу інформації. Це:

- ✓ пункти управління космічними польотами;
- ✓ військові бази;
- ✓ диспетчерські центри контролю над авіаперельотами;
- ✓ брокерські контори, звідки ведеться спостереження за рухом на ринку валют і цінних паперів.
- ✓ моніторингові кімнати провайдерів чи адміністрування ЦОДів

У світі бізнесу без подібного програмного рішення, яке в автоматичному режимі здійснює обробку що надходять відомостей і виводить їх у вигляді зрозумілих діаграм, стовпців, графіків, обійтися складно. Це очевидно. Але «ігри багатих» - далеко не єдина сфера застосування інформаційних панелей. Вони надзвичайно корисні не тільки міжконтинентальних корпораціям, фінансовим мережам, концернам. Працювати з даними за допомогою гарного та інформативного дашборда може будь-хто.

Найпростішим прикладом дашборду, зрозумілим будь-якій сучасній людині, є приладова панель автомобіля. На ній відображено кілька датчиків, які демонструють в зрозумілому вигляді важливу інформацію - швидкість, рівень масла і бензину, число обертів двигуна, температуру в салоні і «за

бортом». Ефективність використання панелі досягається за рахунок групування та мінімізації інформації, що відображається в реальному часі.

Другий приклад елементу дашборду, але вже з галузі моніторингу, це панель на яку виведена кругова діаграма п’яти максимальних навантажень на інтернет лінки, де всі навантаження, менші від певного порогу виведені одним значенням. На панелі буде лише 6 елементів, що дозволить легко сприйняти інформацію.

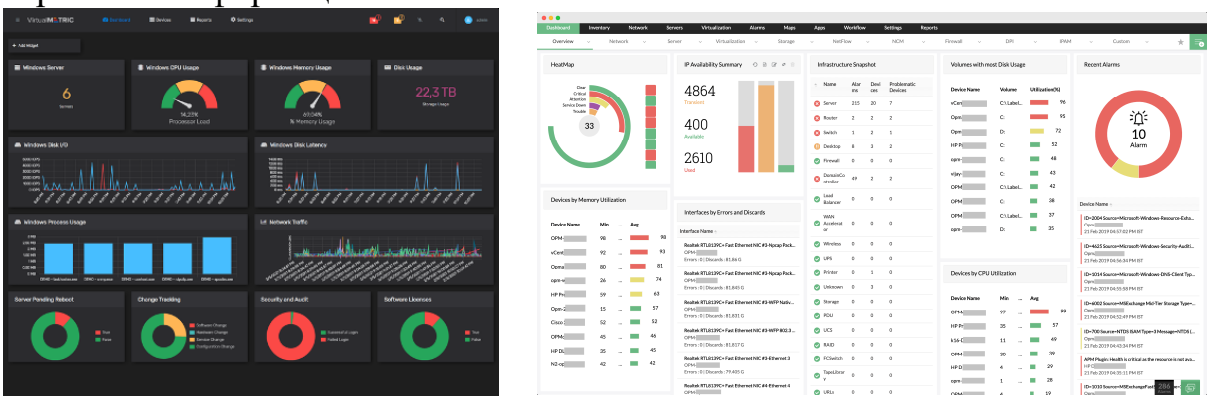


Рис. 6.15. Дашборди моніторингу продуктивності сервера та доступності мережі

Дашборд моніторингу продуктивності сервера (Рис. 6.15). Показує ключові показники продуктивності сервера, такі як використання ЦП, використання пам’яті, дисковий ввід-вивід і мережний трафік. Цю інформацію можна використовувати для виявлення проблем із продуктивністю сервера та їх усунення.

Дашборд моніторингу доступності мережі. Показує доступність мережних пристроїв і зв’язків. Цю інформацію можна використовувати для виявлення проблем із доступністю мережі та їх усунення.

Дашборд моніторингу безпеки мережі. Показує активність мережної безпеки, такі як спроби вторгнення та атаки шкідливих програм (Рис. 6.16). Цю інформацію можна використовувати для виявлення та реагування на загрози безпеці мережі.

Дашборд моніторингу продуктивності веб-сайту. Показує ключові показники продуктивності веб-сайту, такі як час завантаження сторінки, частота відмов і кількість переглядів сторінок (Рис. 6.16). Цю інформацію можна використовувати для виявлення проблем із продуктивністю веб-сайту та їх усунення.

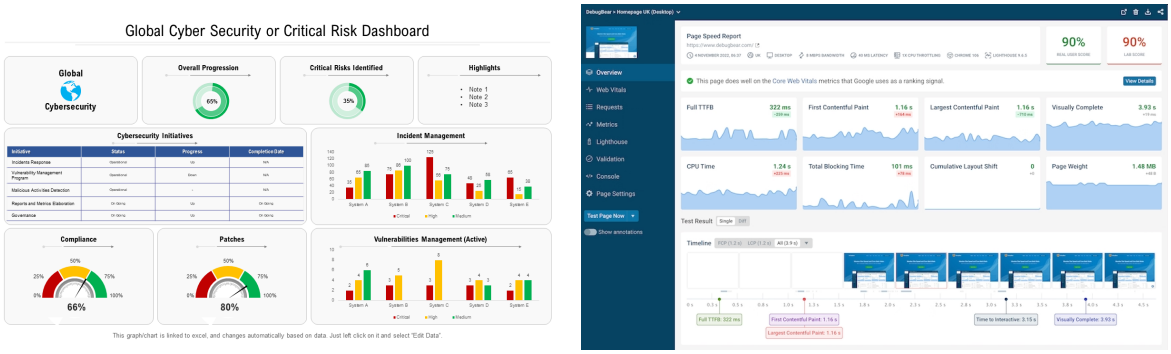


Рис. 6.16. Дашборди моніторингу безпеки мережі та продуктивності веб-сайту

Кожен з нас хоч раз користувався примітивним «родичем» дашборда. Найпростіші зразки - звіт в Excel, рахунок в магазині або кафе, зошит із записом витрат, звіт про рух складу. Від усіх цих прикладів дашборд відрізняється динамічністю, інтерактивністю візуалізації даних.

Візуальні особливості дашборду – зрозумілість і читабельність аналізованих даних досягається в дашборді різноманітними інструментами. Це можуть бути стрілочні індикатори як в транспортних засобах або більш звичні таблиці. Зручний вид анімації програмується під завдання замовника і з урахуванням його особистих побажань.

Будь-яка інформація зі змінними даними може стати джерелом для використання в дашборді - найпростіші таблички в Excel, статистика соцмережі, відвідуваності сайтів, спеціалізовані бази Data Warehouse, багатовимірні куби OLAP і MDX.

Які дії з даними здійснюються в дашборді [5]

- ✓ **Групування.** Схожі відомості групуються за спільною ознакою - назвою, типом, видом дії, стандартністю події.
- ✓ **Агрегація.** Дані відображаються за фактом з вихідних даних, наприклад, сума, кількість, максимум. Ще більш зрозумілий приклад: унікальне число нових відвідувачів сайту за конкретний період часу.
- ✓ **Сортування.** Наприклад, відомості, які колись уже згруповані за конкретним запитом (середнє навантаження на CPU хоста гіпервізора) можна сортувати по висхідній для пошуку хосту з найбільшим навантаженням за період.
- ✓ **Фільтрація.** Відомості виключаються з візуального інструменту за потрібною ознакою або формулою.
- ✓ **Обчислюється колонка.** Нова інформація візуалізується за певними формулами. Наприклад, кількість трафіку за місяць по лінкам провайдера та його тариф дає рахунок, який ми маємо отримати.
- ✓ **Кращі параметри.** На дисплеї відображається число максимальних або мінімальних даних в групі.

Ця та інша інформація відображається на дисплеї у вигляді таблиць, діаграм, карт. Якщо Ви працювали з SQL (Structured query language — мова

структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД), то помітили схожість дій у дашборді та типових SQL-запитів.

Отже, чому для успішної роботи, контролю, аналізу у системі моніторингу потрібна така помічниця як інформаційна панель? Вона дозволяє [36]:

- ✓ систематизувати і візуалізувати дані;
- ✓ представляти їх споживачеві у вигляді простої і наочної анімації;
- ✓ отримувати найостанніші зведення, оперативно спостерігати динаміку даних;
- ✓ видавати дані в потрібній ієрархії, проводити їх порівняння;
- ✓ виділити ключові відомості, щоб тримати їх під контролем;
- ✓ оптимізувати вивчення великих масивів даних, виділяти найважливіше за принципом: «з першого погляду ситуація прояснилася».
- ✓ порівнювати блоки інформації, швидко отримувати відомості в налаштованому для себе вигляді.

Використовуючи дашборди можна отримати [19], [10]

- ✓ Регулярну аналітику за заданими показниками. Такі звіти дозволять відстежити тренди, отримати звіт-аналіз по ситуації і на підставі цих графіків скорегувати тактику компанії.
- ✓ Оперативні звіти - це актуальна інформація, що подається споживачу постійно. Діаграма допоможе не пропустити важливі зміни даних.
- ✓ Довгострокові управлінські рішення складно приймати, не маючи «під рукою» стратегічні звіти. Дашборда допоможе у вирішенні цього завдання.

Як створити дашборд?

Розробка персонального дашборду може коштувати дуже дорого. Набагато вигідніше купити готове програмне рішення, яке легко налаштовується на нового користувача. Існує багато сервісів для розробки таких рішень. Але ми звикли робити все самі, тому…

Основні інструменти створення дашбордів.

Інструмент	Опис	Ліцензія
Grafana	Один з найпопулярніших інструментів для створення дашбордів та візуалізації даних з різноманітних джерел, таких як Prometheus, InfluxDB, Graphite, Elasticsearch тощо. Grafana [37] надає широкий вибір графічних елементів та можливостей налаштування.	Відкрита ліцензія Apache 2.0, безкоштовна для використання.

Kibana	Поставляється разом з Elasticsearch (ELK stack) і використовується для аналізу та візуалізації журналів, метрик та інших даних. Він дозволяє створювати різноманітні графіки, таблиці та інші візуальні елементи для моніторингу.	Відкрита ліцензія Apache 2.0, безкоштовна для використання.
Kubernetes Dashboard	Вбудований інструмент для моніторингу та управління Kubernetes-кластерами. Він надає графіки та статистику щодо ресурсів, а також можливість відслідковувати стан роботи різних компонентів кластера.	Відкрита ліцензія Apache 2.0, безкоштовна для використання.
Prometheus Dashboard	Prometheus поставляється зі своїм власним інструментом для візуалізації даних, що дозволяє створювати графіки та графи для метрик, зібраних Prometheus.	Відкрита ліцензія Apache 2.0, безкоштовна для використання.
Splunk	Інструмент для аналізу та моніторингу журналів та даних. Він має широкий функціонал для створення дашбордів, включаючи графіки, таблиці, зведені дані тощо.	Використовує ліцензію, яка базується на обсязі обробленої даних (дані, які імпортуються та оброблюються Splunk).
MS SQL Server Reporting Services (SSRS)	За допомогою SSRS можна створювати різноманітні звіти та дашборди на основі даних, які зберігаються в базі даних, що підтримує SQL джерела даних. SSRS дозволяє створювати динамічні дашборди, які можуть включати графіки, таблиці, карти та інші візуальні елементи для представлення даних. Ці дашборди будуються на основі запитів до БД, а також можуть використовувати попередньо підготовлені власні набори даних.	Вартість ліцензії залежить від БД, що використовується.

І як резюме хочеться відзначити, що дашборд тільки звучить складно. На практиці це зручне рішення, що демонструють:

5. Глобальні дашборди кіберзагроз.

Спостереження за тим, які зміни відбуваються на планеті в масштабах країн та континентів – справжнє джерело натхнення для аналітиків безпеки. Навіть якщо ви не пов'язані з ІБ, але годинами залипали в контурні карти глобальних стратегій або захоплювалися глобусом в центрі управління X-COM, ця добірка інструментів вам сподобається [34].

Сервісів у цій категорії виявилося чимало. Всі вони мають схожу функціональність, але спираються на різні джерела даних, так що немає однієї карти загроз, яка керує всім.

- shadowserver dashboard — команда Shadowserver збирає інформацію про різноманітні загрози, такі як DDoS-атаки, ботнети, сканування портів та виявлення CVE-уразливостей. На картах і діаграмах представлена статистика щодня оновлюється.
- Global ransomware attacks — глобальна карта інцидентів, що агрегує дані щодо атак програм-вимагачів з 2018 року. Тут можна дізнатися, які сектори економіки торкнулася конкретної атаки, типу програми-вимагача і навіть суми викупу. Оновлюється щодня.
- Live Threat Map — тут можна подивитися зведену статистику по кібератаках за останню годину, добу або місяць, а також виділити країни, що найбільш атакуються, топ векторів атак і порти, що найбільш скануються.
- Talos reputation center - дашборд із загальною інформацією про кіберзагрози, створений компанією Talos за підтримки Cisco.
- Cyber Attack Map - топ нанесених на карту серверів-розповсюджувачів спаму та шкідливого ПЗ.
- Sicherheitstacho – дашборд кібератак від Deutsche Telekom, який працює на базі опенсорсної мережі honeypot (дослівно "горщик з медом): T-Pot . Це система, спеціально розроблена для того, щоб бути скомпрометованою хакерами. Її мета - заманити зловмисників та збирати інформацію про їхні методи та інструменти.
- Cyber map - дашборд від європейської компанії HTTPCS, який агрегує дані про кіберзагрози, виявлені на території різних країн. Для перегляду більшості статистики потрібна реєстрація.
- Cyberthreat Heatmap – інформативний атлас кіберзагроз. Дещо вибивається із загального ряду представлених тут ресурсів, проте пропонує велику кількість аналітики по регіонах та «працюючих» за ними APT-угруповань.
- Sophos threat center — геолокує веб-загрози, джерела розповсюдження спаму та шкідливого ПЗ у міру того, як їх виявляють аналітики компанії

Sophos. Сервіс показує поточний рівень небезпеки та дозволяє побачити базову статистику щодо деяких видів загроз.

- ddos-attack-map - Live-карта DDoS-атак з тимчасовою шкалою, фільтрами по країнах та галузям, а також вказівкою сили атак.
- Digital attack map — інтерактивна карта DDoS-атак, створена в результаті співпраці Google Ideas та Arbor Networks . Дозволяє вивчати історичні дані у звітах про збої в конкретний день із розбивкою країнами.
- DDoS Threat Intelligence Map – аналітична карта, яка, як стверджують творці, дозволяє передбачати джерела майбутніх DDoS-атак. Відображає IP-адреси, з яких велися відбиті атаки, виявляє ботнети, які активно використовувалися за останні 24 години. Надає фільтри за категоріями атакуючих агентів.
- Live botnet threats worldwide — показує розташування IP-адрес серверів, що використовуються для керування зараженими пристроями.
- app.any.run - Дашборд відомого сервісу для перевірки підозрілих файлів на віруси. Відображає статистику щодо малварі, виявленої в різних регіонах світу за останні 24 години.
- Threatbutt Internet Hacking Attack Attribution Map - ця карта виділяється тим, що візуалізує не самі погрози, а повідомлення про атаки. Для цього використовується інформація, знайдена у ЗМІ, соціальних мережах та на сайтах ІБ-компаній. Карта працює на опенсорсному фреймворку pewpew (саме з таким звуком з'являються нові повідомлення).

Глобальні дашборди інтернет-інфраструктури, протоколи та сервіси

- Submarine Cable Map – карта підводних інтернет кабелів. Безкоштовний і регулярно оновлюваний ресурс від TeleGeography
- Wigle – карта точок доступу Wi-Fi
- pingdom map – дашборд для моніторингу доступності сайтів, що оновлюється в Live-режимі.
- DomainTools Internet Statistics - агрегована і нанесена на карту Whois-статистика про IP-адреси і місцезнаходження серверів, їх власників, пов'язаних доменах і багато іншого.

Висновки до розділу

Методи аналізу даних мережного моніторингу є ключовими для виявлення аномалій та проблем в мережі. Вони дозволяють оперативно реагувати на потенційні проблеми та забезпечують безперервну роботу мережної інфраструктури.

Типові інструменти мережного моніторингу, такі як Nagios, Zabbix, PRTG та інші, надають широкий функціонал для збору, аналізу та візуалізації даних про стан мережних пристроїв.

Використання інструментів візуалізації даних є важливим аспектом для розуміння та аналізу стану мережі. Створення графіків, діаграм та звітів дозволяє швидко виявляти та аналізувати тренди та аномалії.

Розробка звітів та дашбордів є ефективним способом візуалізації даних для моніторингу мережі. Ці інструменти дозволяють збирати ключову інформацію та представляти її у зручному та зрозумілому форматі для прийняття обґрунтованих рішень щодо управління мережею.

У підсумку, аналіз та візуалізація даних мережного моніторингу є невід'ємною частиною ефективного управління мережею. Ці процеси допомагають забезпечувати стабільну та безперебійну роботу мережі, вчасно виявляти проблеми та оптимізувати її роботу.

Контрольні запитання

1) Які методи аналізу мережних даних використовують для виявлення аномалій?
2) Що таке статистичний аналіз у мережному моніторингу?
3) Для чого використовується мережне моделювання (Network Modeling)?
4) Як за допомогою графічного аналізу можна виявити активність вузлів мережі?
5) Що таке історичний аналіз продуктивності мережі?
6) Яке призначення має відображення топології мережі (Network topology display)?
7) Що таке перехоплення пакетів (packet sniffing) у мережах?
8) У чому полягає протокольний аналіз (Protocol Analysis)?
9) Яке призначення систем виявлення вторгнень (Intrusion Detection)?
10) Наведіть приклади інструментів аналізатора мережних пакетів.
11) Які проблеми мережі можуть бути пов’язані з продуктивністю?
12) У чому полягає проблема з доступністю мережі?
13) Які основні методи перехоплення пакетів у дротових мережах?
14) Що таке port mirroring і де він застосовується?

Розділ 7. Впровадження систем моніторингу та безпеки IT-інфраструктури

1. Огляд Wazuh як відкритої платформи для моніторингу безпеки

Wazuh – відкрита платформа для моніторингу безпеки, яка поєднує в собі можливості ELK Stack та розширення Wazuh до ELK Stack для виявлення загроз безпеки та реагування на них. Нагадаю, що ELK Stack це комплект відкритих програмних рішень Elasticsearch, Logstash та Kibana [55].

Wazuh починався як проект, заснованої у 2015 році однойменної компанії, що розширює функціональність ELK Stack, додавши можливості моніторингу безпеки. В основі Wazuh лежить інтеграція з ELK Stack, що дозволяє збирати, аналізувати та візуалізувати журнальні дані безпеки.

З часом, розвиток проекту Wazuh призвів до того, що він став самостійною платформою для моніторингу безпеки, яка включає в себе не тільки ELK Stack, але й інші компоненти та функціонал. Таким чином, Wazuh не просто розширює ELK Stack, але створює власну платформу, спеціалізовану на моніторингу безпеки. Вона використовується для виявлення та реагування на загрози безпеки в IT-інфраструктурі.

Основні версії Wazuh:

- ✓ Wazuh 2.x: Випущений близько 2017 року.
- ✓ Wazuh 3.x: Був випущений приблизно в 2018 році.
- ✓ Wazuh 4.x: Остання версія, яка була випущена приблизно у 2020 році.

Ці роки є лише наближеними і можуть відрізнятися залежно від конкретних випусків підверсій та точних дат. На момент написання цього документу актуальна версія Wazuh – 4.12.

Платформа Wazuh надає функції захисту робочих навантажень хмари, контейнера та серверу за допомогою двох важливих інструментів кібербезпеки:

XDR (Extended Detection and Response):

- ✓ Розширене виявлення та реагування: XDR – це комплексна стратегія кібербезпеки, яка використовує дані з різних джерел для виявлення та реагування на кіберзагрози.
- ✓ Консолідація даних: XDR об'єднує дані з телеметрії кінцевих точок, мережного трафіку, журналів та інших джерел для отримання цілісного уявлення про кібербезпеку організації.
- ✓ Автоматизація: XDR використовує машинне навчання та інші методи автоматизації для пришвидшення виявлення та реагування на кіберзагрози.

SIEM (Security Information and Event Management):

- ✓ Управління інформацією та подіями безпеки: SIEM – це програмний продукт, який збирає та аналізує журнали безпеки та інші дані з різних джерел для виявлення кіберзагроз.
- ✓ Кореляція подій: SIEM корелює події з різних джерел, щоб ідентифікувати потенційні кіберзагрози, які інакше могли б залишитися непоміченими.
- ✓ Моніторинг та оповіщення: SIEM використовується для моніторингу мережі та систем на наявність ознак кібератак та для надсилання оповіщень про виявлені загрози.

XDR та SIEM доповнюють один одного. XDR фокусується на виявленні та реагуванні, а SIEM – на зборі та аналізі даних. Використання XDR та SIEM разом може значно покращити кібербезпеку організації.

Переваги Wazuh:

- **Відкритість:** Wazuh - це проект з відкритим кодом, що дає користувачам доступ до його коду та можливість його модифікації.
- **Безкоштовність:** Wazuh можна використовувати безкоштовно, що робить його доступним для організацій з обмеженим бюджетом.
- **Гнучкість:** Wazuh може використовуватися для моніторингу різних типів систем, включаючи сервери, робочі станції, мережні пристрої та хмарні середовища.
- **Кросплатформенність**: Є кросплатформеним рішенням, що означає, що воно підтримується та може бути встановлене на різних операційних системах. Основні операційні системи, на яких можна встановити Wazuh, включають:
  - ✓ **Linux**: Wazuh підтримує багато дистрибутивів Linux, таких як Ubuntu, CentOS, Debian, Fedora, Red Hat Enterprise Linux (RHEL), openSUSE тощо.
  - ✓ **Windows**: Wazuh також може бути встановлено на операційних системах Windows, що дозволяє користувачам моніторити безпеку в їхніх середовищах Windows.
  - ✓ **macOS**: Версії Wazuh також доступні для встановлення на комп'ютери з операційною системою macOS, що дозволяє користувачам забезпечити безпеку своїх систем Mac.

Кросплатформеність Wazuh робить його дуже гнучким і придатним для різних установок та інфраструктур. Це означає, що незалежно від того, яка операційна система використовується в вашій організації, ви можете встановити та використовувати Wazuh для моніторингу та забезпечення безпеки вашої інфраструктури.

- **Масштабованість:** Wazuh може масштабуватися для моніторингу великих IT-інфраструктур.
- **Простота використання:** Wazuh має інтуїтивно зрозумілий інтерфейс користувача, що робить його доступним для користувачів з різним досвідом.

Компоненти Wazuh

Рішення Wazuh базується на агентах Wazuh, які розгортається на контрольованих кінцевих точках, і на трьох центральних компонентах:

- Сервері – Wazuh Server
- Індексаторі – Wazuh indexer
- Інформаційній панелі – Wazuh dashboard

Wazuh Indexer

Індексатор Wazuh (Wazuh indexer) — це система повнотекстового пошуку та аналітики даних безпеки в реальному часі. Дані журналу, що надходять на сервер Wazuh, аналізуються та пересилаються до індексатора для індексування та зберігання. Ці події потім запитуються на інформаційній панелі Wazuh.

Індексатор Wazuh зберігає дані як документи JSON. Кожен документ пов'язує набір ключів, імен полів або атрибутів із відповідними значеннями, якими можуть бути символи, числа, логічні значення, дати, масиви значень, геолокації чи інші види даних.

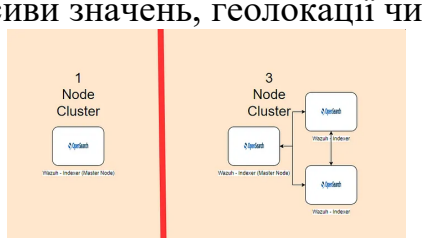


Рис. 7.01. Кластер з 1 вузлом і 3 вузлами

Індексатор Wazuh можна налаштувати як одновузловий або багатовузловий кластер (рис.7.01), що забезпечує масштабованість і високу доступність. Він розподіляє документи між різними контейнерами, відомими як сегменти. У свою чергу, він розподіляє ці фрагменти між вузлами кластера. Розподіляючи документи між декількома шардами та розподіляючи ці сегменти між кількома вузлами, індексатор Wazuh забезпечує надлишковість. Надлишковість забезпечує доступність індексатора Wazuh у разі збою та збільшує пропускну здатність для запитів між вузлами кластера.

Індекси індексатора Wazuh. Покажчик — це сукупність документів, які пов'язані між собою. Індексатор Wazuh використовує індекси для зберігання та організації даних безпеки для швидкого пошуку. Wazuh використовує такі шаблони індексів для зберігання цих даних:

- wazuh-alerts-* : це шаблон індексу для сповіщень, створених сервером Wazuh.
- wazuh-archives-* : це шаблон індексу для всіх подій, надісланих на сервер Wazuh.
- wazuh-monitoring-* : це шаблон індексу для статусу агентів Wazuh.
- wazuh-statistics-* : це шаблон індексу для статистичної інформації сервера Wazuh.

Система підтримує створення спеціальних шаблонів індексу або зміну стандартного шаблону індексу.

Інформація про індекси Wazuhможе бути перевірена двома способами.

- З використанням веб-інтерфейсу користувача.
- Створенням запиту до API індексатора Wazuh.

Переіндексація. Коли в схему даних вносяться зміни, виникає необхідність повторно індексувати дані, щоб відобразити ці зміни. Існуючі дані можуть не відповідати оновленій схемі без повторного індексування, що призведе до невідповідності даних або помилок під час запитів.

Повторне індексування дає змогу копіювати всі або частину ваших даних із вихідного індексу в індекс призначення.

Стандарти безпеки вимагають зберігати дані доступними для аудиту протягом мінімального періоду часу. Дані, які зберігаються довше цього терміну, можуть бути видалені, щоб заощадити місце для зберігання.

Також можливо визначити спеціальні політики для автоматичного видалення даних.

Wazuh Server

Сервер Wazuh аналізує дані, отримані від агентів. Він обробляє їх через декодери та правила, використовуючи аналіз загроз для пошуку добре відомих індикаторів компрометації (IOC). Один сервер може аналізувати дані від сотень або тисяч агентів і масштабувати горизонтально, якщо налаштувати його як кластер. Цей центральний компонент також використовується для керування агентами, налаштування та оновлення їх віддалено, коли це необхідно.

Сервер Wazuh складається з кількох перелічених нижче компонентів (рис.7.02), які мають різний функціонал, наприклад реєстрацію нових агентів, перевірку ідентифікації кожного агента та шифрування зв'язку між агентом Wazuh і сервером Wazuh.

- **Служба реєстрації агентів** використовується для реєстрації нових агентів. Служба надає та розповсюджує унікальні ключі автентифікації кожному агенту. Процес працює як мережна служба та підтримує автентифікацію за допомогою сертифікатів TLS/SSL або шляхом надання фіксованого пароля.
- **Служба підключення агентів** отримує дані від агентів. Служба використовує ключі, спільні для служби реєстрації, для перевірки ідентифікації кожного агента та шифрування зв'язку між агентом Wazuh і сервером Wazuh. Крім того, служба підключення агентів забезпечує централізоване керування конфігурацією, що дозволяє віддалено надсилати нові параметри агентам.
- **Механізм аналізу** це - серверний компонент, який виконує аналіз даних. Він використовує декодери для визначення типу інформації, що обробляється (події Windows, журнали SSH, журнали веб-сервера та інші). Ці декодери також «витягують» відповідні елементи даних із повідомлень журналу, такі як IP-адреса джерела, ідентифікатор події або ім'я користувача. Потім, використовуючи правила, механізм визначає конкретні шаблони в розшифрованих подіях, які можуть ініціювати сповіщення та, можливо, навіть викликати автоматичні контрзаходи (наприклад, блокування IP-адреси, зупинка запущеного процесу або видалення артефакту зловмисного програмного забезпечення).
- **Wazuh RESTful API**. Необхідно розпочати з значення абревіатури назви компоненту. RESTful API розшифровується як Representational State

Transfer Application Programming Interface. REST - це архітектурний стиль для веб-сервісів, API - це інтерфейс програмування, який дозволяє двом програмам взаємодіяти одна з одною. Таким чином, RESTful API - це API, який використовує архітектурний стиль REST для надання доступу до ресурсів. RESTful API надає інтерфейс для взаємодії з інфраструктурою Wazuh. Він використовується для керування параметрами конфігурації агентів і серверів, моніторингу стану інфраструктури та загального стану здоров'я, керування та редагування декодерів і правил Wazuh, а також запиту про стан контрольованих кінцевих точок. Інформаційна панель Wazuh також використовує його.

- **Wazuh cluster daemon** використовується для горизонтального масштабування серверів Wazuh, розгортаючи їх як кластер. Така конфігурація в поєднанні з балансувальником мережного навантаження забезпечує високу доступність і балансування навантаження. Демон кластера Wazuh — це те, що сервери Wazuh використовують для спілкування один з одним і підтримки синхронізації.
- **Filebeat** використовується для надсилання подій і сповіщень до індексатора Wazuh. Він зчитує вихід аналітичної системи Wazuh і передає події в реальному часі а також забезпечує балансування навантаження при підключенні до багатовузлового кластера індексатора Wazuh.

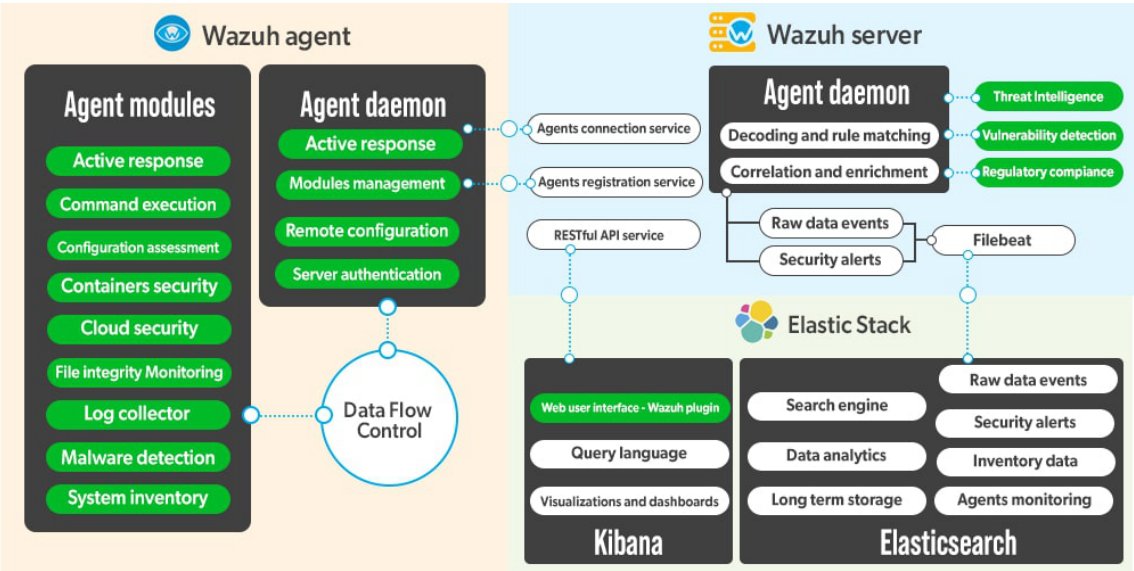


Рис. 7.02. Wazuh Server та агенти у структур Wazuh

Wazuh dashboard

Інформаційна панель Wazuh — це веб-інтерфейс користувача для візуалізації та аналізу даних. Він включає в себе готові інформаційні панелі для

- ✓ подій безпеки
- ✓ нормативної відповідності
- ✓ результатів оцінки конфігурації

- ✓ виявлених вразливостей програм
- ✓ даних моніторингу цілісності файлів
- ✓ моніторингу хмарної інфраструктури
- ✓ подій та ін.

Інформаційна панель Wazuh також використовується для керування конфігурацією Wazuh і моніторингу її стану.

Агенти Wazuh встановлюються на кінцевих точках, таких як ноутбуки, настільні комп'ютери, сервери, хмарні екземпляри або віртуальні машини. Вони забезпечують запобігання загрозам, їх виявлення та реагування. Вони працюють на таких операційних системах, як Linux, Windows, macOS, Solaris, AIX і HP-UX.

На додаток до можливостей моніторингу на основі агентів, платформа Wazuh може контролювати безагентні пристрої, такі як брандмауери, комутатори, маршрутизатори або мережі IDS, серед іншого. Наприклад, дані системного журналу можна збирати через Syslog, а його конфігурацію можна відстежувати шляхом періодичного тестування даних, через SSH або через API.

На діаграмі, що наведена на рисунку 7.03, детально проілюстровано основні компоненти системи Wazuh, а також взаємозв'язки між ними у вигляді потоків даних. Зокрема, на схемі відображено, як окремі модулі системи взаємодіють між собою, передаючи інформацію про події безпеки, журнали та результати аналізу. Діаграма дозволяє краще зрозуміти архітектуру рішення, включаючи розташування агентів, серверної частини та інструментів для обробки й візуалізації даних. Крім того, на ній показано напрямки руху даних — від джерел їх генерації до компонентів, що здійснюють їх збір, аналіз і зберігання. Такий візуальний підхід допомагає більш повно усвідомити принципи функціонування Wazuh, а також роль кожного елементу в загальній системі забезпечення інформаційної безпеки.

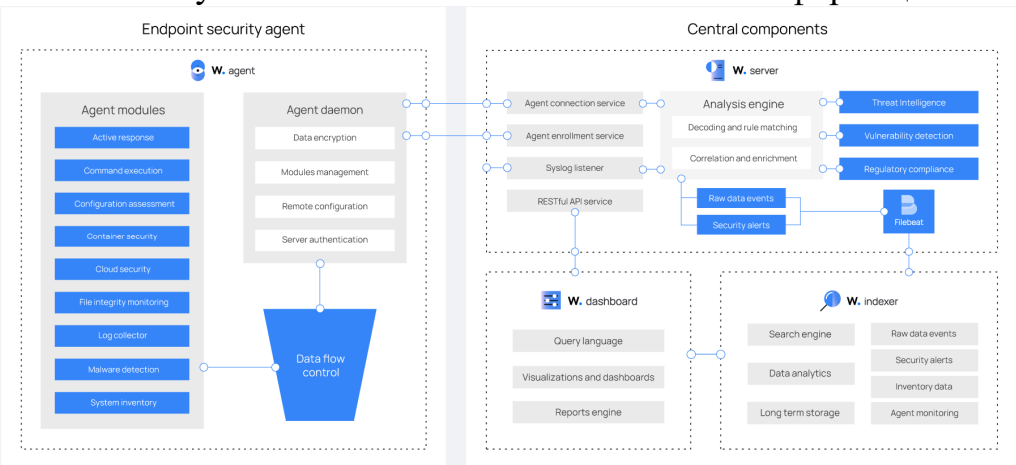


Рис. 7.03. Компоненти та потоки даних Wazuh

Архітектура Wazuh

Архітектура Wazuh базується на агентах , які працюють на контрольованих кінцевих точках і передають дані безпеки на центральний сервер . Безагентні пристрої, такі як брандмауери, комутатори, маршрутизатори та точки доступу, підтримуються та можуть активно надсилати дані журналу через Syslog, SSH або за допомогою свого API. Центральний сервер декодує та аналізує вхідну інформацію та передає результати в індексатор Wazuh для індексування та зберігання.

Кластер індексатора Wazuh — це набір з одного або кількох вузлів, які спілкуються один з одним для виконання операцій читання та запису в індексах. Невеликі розгортання Wazuh, які не вимагають обробки великих обсягів даних, можуть бути легко оброблені кластером з одним вузлом. Багатовузлові кластери рекомендуються, якщо є багато контрольованих кінцевих точок, коли очікується великий обсяг даних або коли потрібна висока доступність.

Кластеризація серверів Wazuh це цікавий та сучасний приклад реалізації масштабування. Реалізація можлива починаючи з версії Wazuh 3.x (2018 рік). По суті, може бути кілька серверів, які працюють разом у режимі кластера та містять протокол, що дозволяє їм обмінюватися інформацією, необхідною для керування підключенням агентів. Іншими словами, агенти зможуть звітувати перед будь-яким нодом (сервером) у кластері, який розподілятиме навантаження між різними вузлами (нодами) та забезпечуватиме можливості високої доступності.

Як показано на рис 7.04, архітектура кластера базується на головному/клієнтському серверах. Головні вузли будуть відповідати за централізацію всієї конфігурації та керування Wazuh.

У рамках кластеризації можливе групування агентів, щоб налаштувати конкретну конфігурацію, політику кореневої перевірки та перевірку надійності для певної групи. Для кожної групи, сервер дистанційно надсилатиме агентам відповідні файли, автоматично застосовуючи зміни.

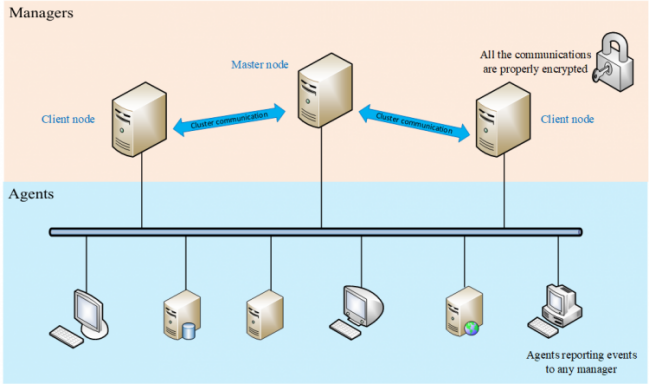


Рис. 7.04. Кластерізація менеджерів Wazuh

Для робочих середовищ рекомендується розгортати сервер Wazuh та індексатор Wazuh на різних хостах. У цьому сценарії Filebeat

використовується для безпечного пересилання сповіщень Wazuh і заархівованих подій до кластера індексатора Wazuh (з одним або кількома вузлами) за допомогою шифрування TLS.

На рис.7.05 показано архітектуру розгортання Wazuh. Він показує компоненти рішення та те, як сервер Wazuh і вузли індексатора Wazuh можуть бути налаштовані як кластери, забезпечуючи балансування навантаження та високу доступність.

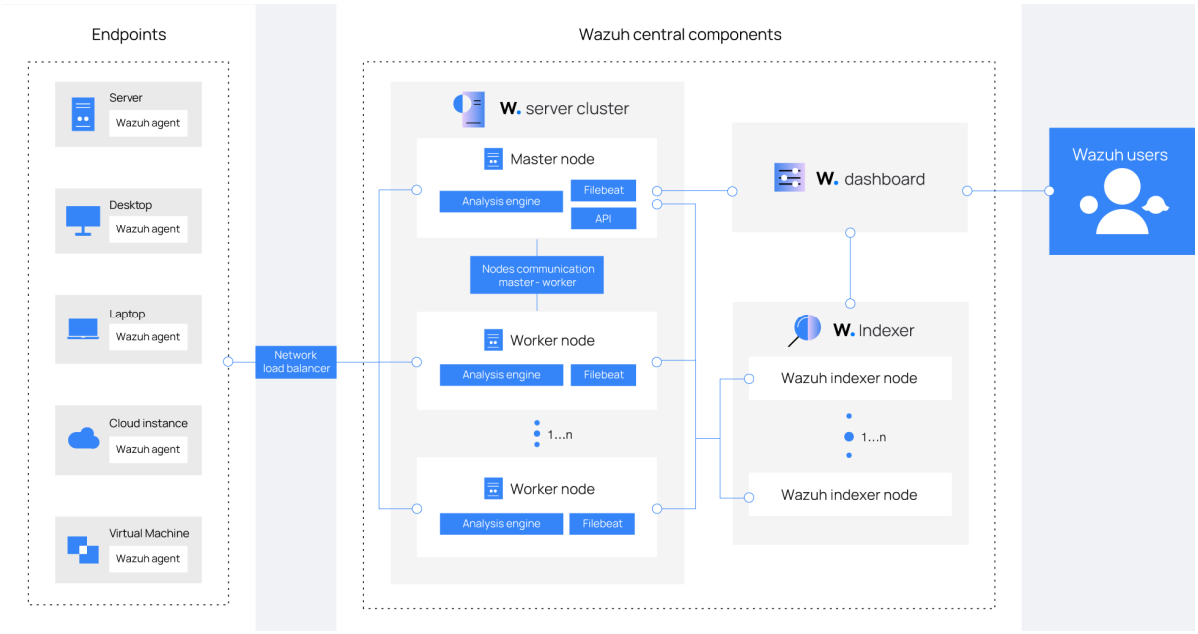


Рис. 7.05. Архітектура розгортання Wazuh

Агент Wazuh постійно надсилає події на сервер Wazuh для аналізу та виявлення загроз. Щоб розпочати надсилання цих даних, агент встановлює з'єднання зі службою сервера для підключення агента, який прослуховує порт 1514 за замовчуванням (номер порту можливо змінити налаштуванням). Потім сервер Wazuh декодує та перевіряє за правилами отримані події, використовуючи механізм аналізу. Події, які запускають правило, доповнюються даними попередження, такими як ідентифікатор правила та назва правила. Події можуть бути передані в один або два файли журналів, в залежності від того, чи спрацьовує правило:

- /var/ossec/logs/archives/archives.json містить усі події незалежно від того, чи спрацювало вони правило чи ні.
- /var/ossec/logs/alerts/alerts.json містить лише події, які спрацювали за правилом із достатньо високим пріоритетом (поріг можна налаштувати).

Протокол повідомлень Wazuh за замовчуванням використовує шифрування AES 128 біт на блок і 256-бітними ключами. Шифрування Blowfish необов'язкове.

Сервер Wazuh використовує Filebeat для надсилання даних попереджень і подій до індексатора Wazuh за допомогою шифрування TLS.

Filebeat зчитує вихідні дані сервера Wazuh і надсилає їх до індексатора Wazuh (за замовчуванням прослуховує порт 9200/TCP). Коли дані проіндексовано індексатором Wazuh, інформаційна панель Wazuh використовується для аналізу та візуалізації інформації.

Інформаційна панель Wazuh запитує Wazuh RESTful API (за замовчуванням прослуховує порт 55000/TCP на сервері Wazuh), щоб відобразити конфігурацію та інформацію про стан сервера та агентів Wazuh . Він також може змінювати параметри конфігурації агентів або сервера через виклики API. Цей зв'язок шифрується за допомогою TLS і автентифікується за допомогою імені користувача та пароля.

Кілька служб використовуються для зв'язку компонентів Wazuh. Не будемо детально зупинятися на списку стандартних портів, які використовуються цими службами. При потребі їх можна розглянути у документації.

Як сповіщення, так і події, що не стосуються сповіщень, зберігаються у файлах на сервері Wazuh, а також надсилаються до індексатора Wazuh. Ці файли можуть бути записані у форматі JSON (.json) або звичайному текстовому форматі (.log). Файли щодня стискаються та підписуються за допомогою контрольних сум MD5, SHA1 і SHA256. Структура каталогу та імені файлу наступна:

```
root@wazuh-manager:/var/ossec/logs/archives/2024/Mar# ls -l
total 176
-rw-r----- 1 wazuh wazuh 234350 Mar  2 00:00 ossec-archive-01.json.gz
-rw-r----- 1 wazuh wazuh    350 Mar  2 00:00 ossec-archive-01.json.sum
-rw-r----- 1 wazuh wazuh 176221 Mar  2 00:00 ossec-archive-01.log.gz
-rw-r----- 1 wazuh wazuh    346 Mar  2 00:00 ossec-archive-01.log.sum
-rw-r----- 1 wazuh wazuh 224320 Mar  2 00:00 ossec-archive-02.json.gz
-rw-r----- 1 wazuh wazuh    350 Mar  2 00:00 ossec-archive-02.json.sum
-rw-r----- 1 wazuh wazuh 151642 Mar  2 00:00 ossec-archive-02.log.gz
-rw-r----- 1 wazuh wazuh    346 Mar  2 00:00 ossec-archive-02.log.sum
```

Рекомендується ротація та резервне копіювання архівних файлів відповідно до обсягу пам'яті сервера Wazuh . Використовуючи завдання cron, легко налаштовується зберігання лише певного часового проміжку архівних файлів локально на сервері, наприклад, минулого року чи останніх трьох місяців.

З іншого боку, можна відмовитися від зберігання архівних файлів і просто покластися на індексатор Wazuh для зберігання архіву. Ця альтернатива може бути кращою, якщо періодично виконується резервне копіювання моментальних знімків індексатора Wazuh або є багатовузловий кластер індексатора Wazuh з копіями фрагментів для високої доступності. Для цього також можна використовувати завдання cron, щоб перемістити знімки індексів на кінцевий сервер зберігання даних і підписати їх за допомогою алгоритмів хешування.

Можливості Wazuh

- **Моніторинг цілісності файлів:** Wazuh може відстежувати зміни файлів та повідомляти про підозрілу активність.
- **Моніторинг журналів:** Wazuh може збирати та аналізувати журнали з різних систем для виявлення загроз безпеки.
- **Моніторинг мережі:** Wazuh може відстежувати мережний трафік та повідомляти про підозрілу активність.
- **Виявлення вторгнень:** Wazuh може використовувати правила Snort для виявлення вторгнень.
- **Моніторинг аномалій:** Wazuh може використовувати машинне навчання для виявлення аномальної поведінки.
- **Реагування на інциденти:** Wazuh може автоматизувати дії реагування на інциденти.

Wazuh – це система виявлення вторгнень (IDS) та система запобігання вторгненням (IPS), а не всебічний інструмент моніторингу та кібербезпеки.

Треба розуміти, що Wazuh не має вбудованих функцій для моніторингу продуктивності та веб-додатків, не є антивірусом, не має вбудованих функцій для шифрування даних, не може моніторити хмарні інфраструктури без додаткових плагінів або інтеграцій, не може автоматично виправляти вразливості, не має вбудованих функцій для моніторингу IoT-пристроїв та не має вбудованих функцій для моніторингу контейнерів та баз даних.

Використання Wazuh для моніторингу цілісності файлів

Для цієї задачі Wazuh використовує модуль File Integrity Monitoring (FIM), що відстежує та сповіщає про зміни критичних файлів і каталогів та швидко виявляти зміни файлів, які вказують на компрометацію чи кібератаку. Використання FIM надає Wazuh наступні особливості:

- **Моніторинг в реальному часі** – виявлення змін файлів і реакція на них у реальному часі (рис.7.06). Wazuh відстежує системні файли та каталоги в режимі реального часу, щоб виявляти зміни, коли вони відбуваються, і запускає сповіщення, які дозволяють негайно вжити заходів. Це допомагає організаціям пом’якшити вплив інцидентів безпеки
- **Виявлення порушень безпеки та втручань в систему за допомогою Wazuh FIM.** FIM (File Integrity Monitoring) - модуль вбудований в Wazuh, що використовується для моніторингу та оповіщення про зміни в критичних файлах та директоріях. Wazuh відстежує файли та каталоги, відстежуючи атрибути, дозволи, право власності та вміст. Він використовує хеш-значення для виявлення змін у файловій системі, виявлення зловмисних дій і зменшення внутрішніх загроз від окремих осіб або постачальників.

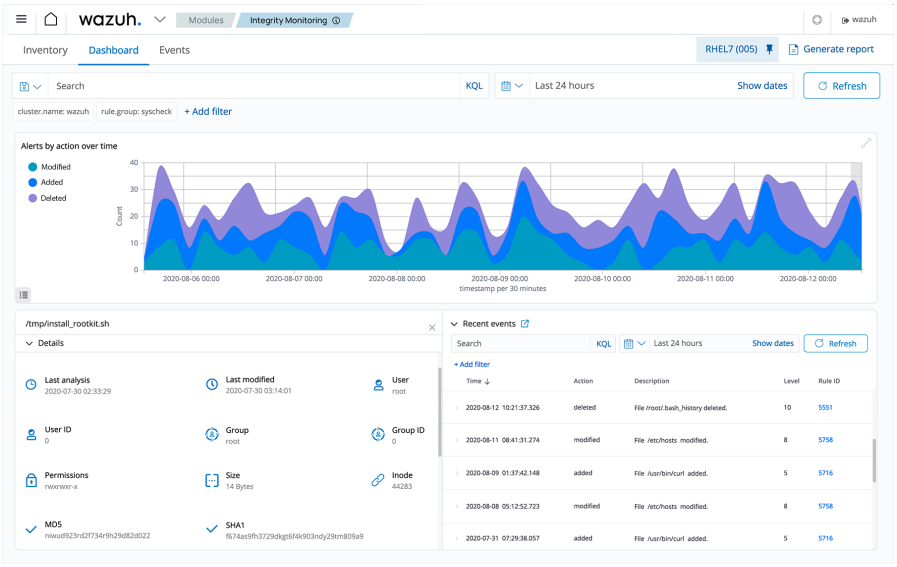


Рис. 7.06. DashBoard Wazuh

Як працює Wazuh FIM:

1. **Сканування**: Модуль FIM періодично сканує певні шляхи та моніторить певні директорії в режимі реального часу. Ви можете встановити, які шляхи моніторити в конфігурації агентів та менеджера Wazuh.
2. **Базова лінія**: FIM створює базову лінію, зберігаючи криптографічну контрольну суму та інші атрибути моніторингових файлів.
3. **Порівняння**: FIM порівнює інформацію базової лінії з інформацією про останню версію файлу. Це порівняння дає видимість змін та оновлень критичних файлів.
4. **Оповіщення**: Якщо FIM виявляє будь-які зміни в моніторингових файлах, він генерує сповіщення, яке може бути відправлене на різні канали, такі як електронна пошта, SNMP або syslog.

Методи встановлення Wazuh

- **Вимоги до обладнання** значною мірою залежать від кількості захищених кінцевих точок і хмарних робочих навантажень. Це число може допомогти оцінити, скільки даних буде проаналізовано та скільки сповіщень безпеки буде збережено та проіндексовано.

Комплекту центральних компонентів (Wazuh server, indexer, dashboard) на одному хості достатньо для моніторингу до 100 кінцевих точок і для 90 днів запитуваних/індексованих даних попереджень. У таблиці нижче показано рекомендована конфігурація для швидкого розгортання:

Агенти	CPU	RAM	Зберігання (90 днів)
1–25	4 vCPU	8 Гб	50 ГБ
25–50	8 vCPU	8 Гб	100 ГБ
50–100	8 vCPU	8 Гб	200 ГБ

Для великих середовищ ми рекомендуємо розподілене розгортання. Конфігурація кластера з кількома вузлами доступна для сервера Wazuh і для індексатора Wazuh, що забезпечує високу доступність і балансування навантаження.

- **Операційна система.** Центральні компоненти Wazuh можна встановити на 64-бітну операційну систему Linux. Wazuh рекомендує будь-яку з наступних версій ОС: Amazon Linux 2, CentOS 7, 8, Red Hat Enterprise Linux 7, 8, 9, Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04
- **Стандартне встановлення Wazuh.** Завантажте та запустіть помічника встановлення Wazuh.

  curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh –a

  Коли помічник завершить інсталяцію, у вихідних даних відобразяться облікові дані доступу та повідомлення, яке підтверджує, що інсталяція пройшла успішно.

  INFO: --- Summary ---
  INFO: You can access the web interface https://<wazuh-dashboard-ip>
  User: admin
  Password: <ADMIN_PASSWORD>
  INFO: Installation finished.

  Wazuh встановлено та налаштовано. Доступ до веб-інтерфейсу Wazuh за допомогою

  https://<wazuh-dashboard-ip>

  Облікові дані:
  Ім'я користувача: admin
  Пароль: <ADMIN_PASSWORD>

Коли доступ до інформаційної панелі Wazuh отримано вперше, браузер показує попередження про те, що сертифікат не був виданий довіреним центром. Це очікувано, і користувач має можливість прийняти сертифікат як виняток або, альтернативно, налаштувати систему на використання сертифікату від довіреного центру.

Можливо розпочати розгортання агентів Wazuh, що використовуються для захисту ноутбуків, настільних ПК, серверів, хмарних примірників, контейнерів або віртуальних машин. Агент легкий і багатоцільовий. Інструкції щодо розгортання агентів Wazuh можна знайти у веб-інтерфейсі користувача Wazuh або в документації.

- **Встановлення Wazuh розгортанням готових до використання машини**
  - ✓ **Віртуальна машина (OVA).** Wazuh надає попередньо зібраний образ віртуальної машини (OVA), який можна імпортувати безпосередньо

за допомогою VirtualBox або інших систем віртуалізації, сумісних з OVA.

- ✓ **Образи машини Amazon (AMI).** Попередньо створений образ машини Amazon (AMI), який можна запускати безпосередньо в хмарному екземплярі AWS.

- ➢ **Встановлення контейнерів.**
  - ✓ **Розгортання на Docker**. Docker — це набір продуктів платформи як послуги (PaaS), які доставляють програмне забезпечення в пакетах, які називаються контейнерами. За допомогою Docker ви можете встановити та налаштувати розгортання Wazuh як архітектуру з одним хостом.
  - ✓ **Розгортання на Kubernetes**. Kubernetes — це система з відкритим кодом для автоматизації розгортання, масштабування та керування контейнерними програмами. Цей тип розгортання використовує зображення Wazuh із Docker і дозволяє створювати середовище Wazuh.

2. Огляд Snort як системи виявлення та запобігання вторгненням

Snort — це одна з найпопулярніших у світі систем виявлення та запобігання вторгненням (IDS/IPS) з відкритим вихідним кодом. Вона забезпечує ефективний аналіз мережного трафіку в режимі реального часу та здатна як виявляти, так і блокувати підозрілу або небезпечну мережну активність [44].

Основою роботи Snort є система правил. Ці правила описують сигнатури зловмисної активності, наприклад, характерні ознаки DoS-атак, спроб сканування портів, експлойтів, аномальних протоколів тощо. Snort порівнює вхідні мережні пакети з цими правилами, і у разі збігу може або згенерувати сповіщення, або відразу заблокувати небезпечний трафік, залежно від обраного режиму роботи.

Snort підтримує три основні режими використання:

- ➢ **Сніффер трафіку** — подібно до утиліти tcpdump, Snort дозволяє переглядати мережні пакети у режимі реального часу.
- ➢ **Логер (реєстратор)** — фіксує трафік у лог-файли для подальшого аналізу чи налагодження.
- ➢ **Повноцінна система запобігання вторгненням (IPS)** — здійснює аналіз трафіку та може блокувати зловмисні пакети відповідно до налаштованих правил.

Snort є доступним для завантаження та використання як у навчальних, так і в корпоративних середовищах. Однією з його переваг є підтримка різних наборів правил:

- Набір правил спільноти (**Community Ruleset**) — створений та підтримуваний активною спільнотою користувачів Snort. Цей набір є безкоштовним і відкритим для всіх охочих.
- Набір правил для передплатників (**Snort Subscriber Ruleset**) — розробляється та перевіряється командою безпеки Cisco Talos. Цей набір призначений для комерційних користувачів, які мають підписку, і отримують оновлення правил у режимі реального часу одразу після їх виходу для клієнтів Cisco.
- **Registered User Ruleset** (безкоштовно, але з затримкою) До цього типу правил ще повернемося, коли торкнемось ліцензування.

Оновлення правил та налаштування Snort здійснюється через офіційний портал Snort.org. Серед ключових переваг Snort можна відзначити:

- **Гнучкість налаштування**. Можливість створювати власні правила для специфічних потреб мережі.
- **Підтримка інтеграції з іншими системами моніторингу**. Як приклад, журнали Snort можуть бути імпортовані до SIEM-рішень таких як Wazuh, Splunk або QRadar для поглибленого аналізу.
- **Широке застосування у корпоративних середовища**х завдяки високій продуктивності, великій спільноті та надійності.

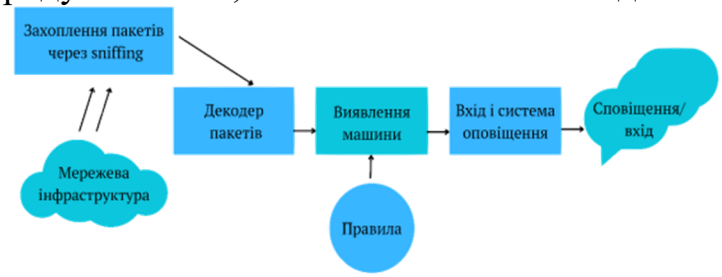


Рис. 7.07. Архітектура Snort

Архітектура Snort має чітко структуровану модульну будову, що зображена рис. 7.07. Основні компоненти:

- **Захоплення пакетів через sniffing.** Snort починає свою роботу з прослуховування мережного трафіку. За допомогою механізму сніффінгу система отримує всі вхідні пакети, що проходять через мережу.
- **Декодер пакетів.** Захоплені пакети надходять на декодер, який розбирає кожен пакет, ізолюючи заголовки протоколів, що відповідають різним рівням моделі OSI. Це дозволяє Snort отримати інформацію про структуру пакета для подальшого аналізу.

- **Виявлення машини (детекція).** Основна функціональність Snort зосереджена саме в цьому блоці. Механізм виявлення аналізує вміст кожного пакета, порівнюючи його з набором правил. Правила можуть визначати характерні ознаки атак, спроб проникнення або аномальної активності в мережі.
- **Система правил.** Серце Snort — це база правил, яка може бути як стандартною (наприклад, набір Community Ruleset), так і кастомізованою під специфіку конкретного середовища. Кожне правило описує умови, за яких трафік вважається підозрілим або зловмисним.
- **Вхід і система оповіщення.** Якщо пакет відповідає умовам певного правила, система спрацьовує: формується запис про подію та надсилається відповідне сповіщення адміністратору або іншим захищеним системам.

Таким чином, Snort функціонує як потужний фільтр мережного трафіку, здатний оперативно реагувати на загрози.

Архітектура Snort демонструє його здатність ефективно інтегруватися в корпоративну мережну інфраструктуру як система первинного аналізу та виявлення загроз. Однак максимальну ефективність Snort досягає, коли виступає не як окремий інструмент, а як джерело подій для більш комплексних рішень.

Варто зазначити, що Snort виконує роль одного з основних джерел подій про мережну активність. Його доцільно використовувати у поєднанні з іншими системами, які ми розглядаємо у цій темі, для побудови комплексної системи моніторингу та безпеки IT-інфраструктури. Наприклад, Snort можна використовувати для детекції атак на мережному рівні, а зібрані події корелювати у SIEM-системах для отримання повної картини безпекових інцидентів.

Snort є важливим елементом в арсеналі засобів забезпечення безпеки корпоративних мереж, що доповнює можливості інших рішень, підсилюючи загальний рівень захисту мережної інфраструктури.

Встановлення Snort. Підтримувані операційні системи. Snort є кросплатформеним рішенням і може бути встановлений на:

- Linux (найпопулярніші дистрибутиви: Ubuntu, Debian, CentOS, Fedora, Red Hat)
- FreeBSD
- macOS

- Windows (є спеціальний інсталятор для Windows, але використовується рідше через складнішу інтеграцію)

Основні залежності. Перед встановленням Snort необхідно підготувати систему, встановивши такі компоненти:

- Libpcap (для Linux) / WinPcap / npcap (для Windows). Бібліотека для захоплення мережного трафіку.
- DAQ (Data Acquisition Library). Бібліотека, яка надає Snort доступ до мережного інтерфейсу для роботи з пакетами.
- PCRE (Perl Compatible Regular Expressions). Використовується для роботи з регулярними виразами у правилах Snort.
- Libdnet (на Linux). Потрібна для низькорівневої роботи з мережею.
- Barnyard2 (необов'язково). Додатковий компонент для обробки та експорту логів Snort у базу даних або SIEM.
- PulledPork (опціонально). Інструмент для автоматичного завантаження та оновлення правил Snort.

Snort має «двоїсте ліцензування», що дозволяє гнучко використовувати його як у комерційних, так і некомерційних цілях:

- **GPLv2 (GNU General Public License, версія 2)**. Це базова ліцензія Snort, яка дозволяє безкоштовне використання, модифікацію та розповсюдження програмного забезпечення. Вона підходить для освітніх цілей, лабораторій, дослідницьких проектів і навіть використання у продуктивному середовищі, якщо дотримано вимоги GPL.
- **Комерційна ліцензія від Cisco.** Оскільки Snort належить компанії Cisco, існує можливість укладання комерційної угоди для інтеграції Snort у пропрієтарні рішення або для отримання додаткової підтримки та розширених можливостей.

Правила Snort з урахуванням ліцензування:

- **Community Rules** (безкоштовні) — доступні всім користувачам Snort.
- **Subscriber Ruleset** (платна підписка) — більш розширений набір правил з оновленнями для комерційних клієнтів та організацій, які бажають отримувати нові правила якомога швидше.
- **Registered User Ruleset** (безкоштовно, але з затримкою) — доступні після реєстрації, але з деякою затримкою порівняно з платною підпискою.

Таким чином, для навчальних цілей та базового використання достатньо Community Rules та GPLv2 ліцензії. Для серйозних корпоративних впроваджень варто розглядати підписку на оновлення правил або укладення комерційного контракту.

3. Огляд Splunk Enterprise Security

Splunk Inc. — це американська компанія -розробник програмного забезпечення , розташована в Сан-Франциско, Каліфорнія, яка виробляє програмне забезпечення для пошуку, моніторингу та аналізу машинно-генерованих даних через веб-інтерфейс. Його програмне забезпечення допомагає отримувати, індексувати та співвідносити дані в реальному часі в репозиторії з можливістю пошуку, з якого можна створювати графіки, звіти, сповіщення, інформаційні панелі та візуалізації [56].

Фірма використовує машинні дані для ідентифікації шаблонів даних, надання показників, діагностики проблем і надання інформації для бізнес-операцій. Це горизонтальна технологія, яка використовується для керування програмами , безпеки та відповідності , а також для бізнес- та веб-аналітики.

У вересні 2023 року було оголошено, що Splunk буде придбана компанією Cisco за 28 мільярдів доларів США за угоду за готівку. Трансакцію було завершено 18 березня 2024 року.

Це була найбільша угода в історії Cisco. На той час у Splunk було 1100 патентів із такими клієнтами, як Singapore Airlines, Papa Johns, Heineken і McLaren. Splunk продовжував працювати під тим самим керівництвом, а ціни залишалися незмінними.

Splunk Enterprise Security (ES) – система управління інформаційною безпекою та подіями (англ. Security and information event management, SIEM), яка формує детальну картину машинних даних, що створили додаткові технології безпеки (мережа, кінцеві точки, доступ, шкідливі програми, вразливість і перевірка). Завдяки Splunk Enterprise Security фахівці з безпеки швидко ідентифікують внутрішні та зовнішні атаки та приймають відповідні запити. Splunk Enterprise Security дозволяє спростити операції із захисту від загроз, мінімізувати ризик і забезпечити безпеку вашого бізнесу. Рішення оптимізує всі аспекти захисту та підходить для організації будь-якого масштабу та професійного рівня, дозволяє здійснювати пошук кореляцій, оповіщення, звітності та панелі моніторингу з урахуванням ваших потреб.

Splunk ES використовується для постійного моніторингу в реальному часі, швидкого реагування на інциденти, в якості операційного захисту центру або для керівників, які необхідні відомості про ділові ризики.

Ключові можливості застосування продукту:

- **Моніторинг в реальному часі** – отримання чіткого і наочного виявлення про стан безпеки організації, крім налаштувань виявлення і деталізації даних аж до базових подій.
- **Призначення пріоритетів і вжиття відповідних заходів** – представлення даних у контексті безпеки для розширення можливості виявлення загроз та оптимізації реагування на інциденти.
- **Швидке розслідування** – використання ситуативного пошуку, а також статичних, динамічних та візуальних кореляцій для виявлення шкідливих дій.
- **Багатоетапні розслідування** – аналіз пошкоджений і наслідковий аналіз для виявлення динамічних дій, пов'язаних із загрозами.

Splunk Enterprise Security запускається в середовищі Splunk Enterprise або Splunk Cloud. Рішення Splunk ES може бути розгорнуто у вигляді програми або хмарної служби, у загальнодоступному або приватному, а також у гібридному програмно-хмарному середовищі.

Ось основні результати впровадження у якості SIEM саме Splunk Enterprise Security:

- **Поліпшення операцій по забезпеченню безпеки.** Прискорене реагування на інциденти та демонстрація відповідності вимогам за допомогою великого обладнання вбудованих панелей моніторингу, видимих таблиць, звітів і процесів реагування на інциденти, включаючи оцінки ризиків, швидкий пошук, аналітику, кореляції та індикатори безпеки.
- **Підвищення рівня безпеки.** Оптимізація процесів моніторингу безпеки, розстановки пріоритетів, реагування, стримування та відновлення аналізу всіх машинних даних для оцінки впливу сповіщень або інцидентів.
- **Призначення пріоритетів подій безпеки та розслідувань**. Поліпшення процесу прийняття рішень і розробка стратегії усунення ризиків з урахуванням вимог бізнесу шляхом застосування оцінок ризиків до будь-якої події, активу, виклику чи користувача у зв'язку з їх відносною важливістю або цінністю для бізнесу.
- **Виявлення внутрішніх і складних загроз.** Перевірка привілейованого доступу та незвичайної діяльності шляхом використання даних про відхилення, виявлених додатком UBA, а також застосування користувальницького контексту та контексту активів до всіх машинних даних для моніторингу користувачів та активів.
- **Ухвалення більш обґрунтованих рішень.** Більш ефективне розслідування інцидентів і активний захист, оцінка їх масштабу на основі даних про загрози з різних джерел, включаючи безкоштовні канали аналізу загроз, підписки на послуги сторонніх постачальників,

правоохоронні органи, FS-ISAC, STIX / TAXII, Facebook ThreatExchange, а також інші внутрішні та загальнодоступні джерела.

- **Використання аналізу загроз.** Аналітичні дані по загрозам з небагатьох джерел можна об'єднувати, дедуплицировать і позначати ними ваги. Це дозволяє використовувати широкий спектр індикаторів компрометації для всіх аспектів моніторингу, оповіщення, звітності, розслідувань і криміналістичного аналізу.
- **Моніторинг в реальному часі.** Виявлення незвичайних дій, які можуть бути ознаками складних загроз, за допомогою статистичного аналізу, відхилень, виявлених додатком UBA, пошуку кореляцій, динамічних граничних умов і засобів виявлення відхилень від норми.
- **Оптимізація реагування на інциденти.** Оптимізація розслідувань динамічних, багатоетапних атак завдяки можливості візуалізувати інформацію про атаки і, отже, краще розуміти через час, а також будувати наступний ланцюжок різних подій для швидкого визначення подальших дій.
- **Підвищення операційної ефективності.** Автоматичне і напівавтоматичне прийняття рішень допомагає клієнтам прискорити розслідування та вжиття заходів з повним контекстом у програмі Adaptive Response.
- **Розуміння значення показників безпеки.** Спрощення аналізу за допомогою логічних або фізичних проявів візуальної таблиці, що дозволяє краще дізнатися про основні показники безпеки, включаючи доступ, DNS, ідентифікацію, електронну пошту, IDS, ліцензування, шкідливі ПЗ, події, продуктивність, ризики, SSL, шкідливі дії, трафік, UBA, оновлення, вразливість і мережну активність.

4. Огляд IBM QRadar: архітектура та можливості

IBM QRadar — це комплексне рішення у сфері управління інформаційною безпекою та подіями (SIEM - Security Information and Event Management), яке використовується для моніторингу, аналізу та реагування на кіберзагрози в корпоративних мережах. QRadar поєднує в собі різні технології аналізу мережного трафіку, журналів подій, управління вразливостями та автоматизації процесів реагування, що робить його одним із найбільш потужних продуктів у сфері кібербезпеки [57].

Перш ніж розглянути QRadar у інфраструктурі, необхідно зрозуміти кілька основних компонентів, які забезпечують його належну роботу.

Модульність

IBM QRadar SIEM (система управління інформацією та подіями безпеки) має модульну архітектуру, що дозволяє масштабувати розгортання

шляхом додавання нових пристроїв, кінцевих точок і серверів до вашої інфраструктури для покращення аналізу та збору логів. Також можна інтегрувати додаткові модулі для аналізу, які легко завантажуються з IBM App Exchange. До них, зокрема, належать:

- ✓ QRadar Vulnerability Manager (менеджер вразливостей QRadar)
- ✓ QRadar Risk Manager (менеджер ризиків QRadar)
- ✓ QRadar Watson Integration (інтеграція з Watson)
- ✓ QRadar Incident Forensics (цифрова криміналістика або розслідування інцидентів інформаційної безпеки QRadar)

Цей список не є вичерпним, оскільки доступні й інші модулі для розширення функціональності.

Три рівні: що це таке?

Архітектура QRadar також використовує концепцію рівневості та складається з трьох основних рівнів, через які інформація передається від кінцевих точок до агента. Зверніть увагу, що ця архітектура застосовується у всіх типах розгортань, незалежно від розміру організації, складності мережі чи кількості використовуваних компонентів.

IBM QRadar у реальному часі збирає, обробляє, агрегує, корелює, зберігає та візуалізує дані з усіх підключених кінцевих точок. Опрацьовані та очищені дані надають вам аналітичну інформацію для моніторингу IT-інфраструктури. Це зазвичай реалізується у вигляді сповіщень, інцидентів та відповідей на загрози, які QRadar виявляє в режимі реального часу.

До розгортання можна також додати додаткові модулі, які ми згадували раніше. Наприклад, модуль Incident Forensics допомагає проаналізувати дії зловмисника, виявити його сліди та швидше зробити висновки на основі залишених артефактів. Це значно скорочує час, необхідний команді реагування на інциденти (IR), для розслідування загроз, зафіксованих QRadar. Він також сприяє швидкому усуненню порушень, запобіганню втрат даних і новим атакам.

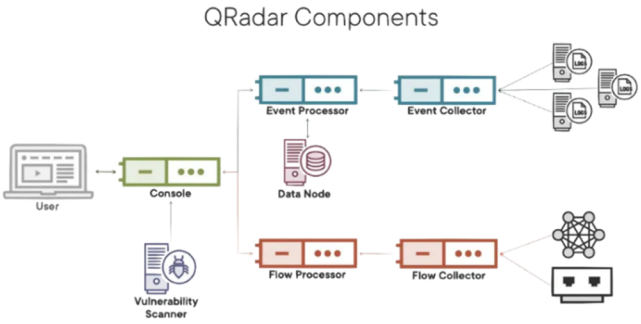


Рис. 7.08. Типова архітектура IBM QRadar

Повертаємося до рівнів. Далі розглянемо, які три рівні містить архітектура QRadar (рис.7.08) і детальніше заглибимося в кожен із них:

- ❖ **Збір даних (Data collection)**
- ❖ **Обробка даних (Data processing)**
- ❖ **Пошук даних (Data searches)**

➢ **Збір даних (Data Collection)** є нижнім або першим рівнем архітектури QRadar, і його місія проста: «збирати все з вашої мережі». «Збір» включає такі типи даних, як:

✓ Журнали подій (events, log files)

✓ Мережні потоки (flows)

✓ Інші типи інформації, такі як результати сканування, конфігураційні файли, знімки трафіку (packet captures) тощо…

Для збору подій і потокових даних QRadar використовує QRadar Event Collectors та QRadar QFlow Collectors (про «потоки» ми поговоримо далі).

QRadar може отримувати події з різних джерел у мережі, таких як фаєрволи, системи запобігання вторгнень (IPS) та інші пристрої безпеки. Для збору даних використовуються різні протоколи, найпоширенішим з яких є syslog. Інші варіанти:

- ✓ syslog-tcp
- ✓ SNMP
- ✓ SCP, SFTP, FTP (для отримання подій)
- ✓ JDBC
- ✓ Check Point OPSEC
- ✓ SMB/CIFS

Два основних типи даних. QRadar працює з двома основними типами даних:

✓ **Дані подій (Event Data).** Події відображають активність, яка відбулася на кінцевій точці у певний момент часу. Наприклад:

- Зміна конфігурації фаєрвола
- Збій у мережі
- Невдалі спроби входу
- Вхідні/вихідні підключення
- Логування електронної пошти тощо

✓ **Дані потоків (Flow Data).** Мережні потоки містять інформацію про сесії зв'язку між пристроями в мережі. Потоки конвертуються у так звані «записи потоку» (flow records), які містять:

- IP-адреси
- Порти
- Обсяг переданих байтів
- Кількість пакетів
- Протокол зв'язку

Хоча цей етап носить назву Data Collection, дані все одно проходять на ньому первісну обробку:

1. **Парсинг (Parsing).** Сирі події перетворюються у зручний для читання формат. Це виконується за допомогою DSM Editor (Device Support Model Editor), що дозволяє швидко адаптувати дані під необхідний формат. Перед налаштуванням парсера необхідно забезпечити отримання журналів через syslog або інші протоколи.

2. **Нормалізація (Normalization).** На цьому етапі дані перетворюються у формат, що містить корисні поля (наприклад, IP-адреси), які QRadar може використовувати для подальшого аналізу.
3. **Визначення джерела подій.** Після нормалізації QRadar ідентифікує джерело подій на основі заголовків IP. Події агрегуються у записи, проте якщо QRadar не може визначити джерело журналів (log source), вони направляються до модуля аналізу трафіку для автоматичного виявлення. Якщо QRadar знаходить нове джерело логів, він надсилає запит на додавання цього джерела у консоль керування QRadar.

Крім цього, за допомогою модуля Incident Forensics можна виконувати повний перехоплення трафіку (packet capture) для глибшого аналізу загроз. Таким чином, на рівні збору подій QRadar виконує основні завдання:

✓ Збирає дані за допомогою відомих протоколів

✓ Моніторить події та керує чергами для оптимального навантаження

✓ Парсить сирі події у формат, придатний для аналізу

✓ Аналізує джерела логів за допомогою DSM, підтримує автоматичне виявлення

✓ Агрегує події на основі спільних атрибутів

✓ Передає події до інших SIEM-систем, якщо це необхідно

Переходимо до наступного рівня — обробки даних (Data Processing).

- **Обробка даних (Data Processing).** Після того як дані пройшли всі необхідні етапи на першому рівні (збору даних), вони передаються на другий рівень — рівень обробки. Тут працюють два основних компоненти:

✓ Обробники подій (Event Processors)

✓ Обробники потоків (Flow Processors)

Їх завдання — обробити отримані дані та передати їх на фінальний рівень. Пройшовши етап збору, дані трансформуються, перш ніж їх можна буде використовувати для пошуку та аналізу. Як відбувається обробка?

1. Перше, через що проходять дані, — це механізм Custom Rules Engine (CRE). Це механізм спеціальних правил, який аналізує вхідні події. Саме тут створюються сповіщення та інциденти (offenses & alerts) на основі отриманих даних. Усі події зберігаються для довготривалого зберігання (storage persistence).
2. Як працюють правила в CRE? Адміністратори налаштовують користувацькі правила (custom rules) через консоль QRadar. Якщо отримані події відповідають умовам правил, спрацьовують певні дії або реакції. Таким чином, QRadar може автоматично виявляти загрози та реагувати на них у реальному часі.

3. Передача подій у консоль в реальному часі Обробники подій (Event Processors) передають дані безпосередньо в консоль QRadar. Вся інформація миттєво відображається у вкладці “Log Activity” (Журнал активності). Дані обробляються в реальному часі, а не відтворюються з бази.
4. Збереження даних. Події не тільки аналізуються в потоці, але й зберігаються для подальшого аналізу та розслідувань. Отже, другий рівень відповідає за аналіз подій, застосування правил безпеки та збереження даних для подальшого використання.

Крім того, у Data Processing є ще один рівень абстракції, який ми раніше не згадували, – це Magistrate. Custom Rules Engine (CRE) не передає співставлені події безпосередньо на консоль. Замість цього вони спочатку надходять до Magistrate на консолі QRadar. Його завдання – створювати та керувати інцидентами (offenses) на консолі. Magistrate виконує три основні функції:

- ✓ Правила для інцидентів (Offense Rules) – визначає, що робити у разі створення інциденту.
- ✓ Керування інцидентами (Offense Management) – оновлення статусу інцидентів, додавання додаткової інформації тощо.
- ✓ Зберігання інцидентів (Offense Storage) – всі події зберігаються у базі даних PostgreSQL.

Окрім цього, Magistrate Processing Core (MPC) відповідає за кореляцію інцидентів з повідомленнями про події, що надходять від різних Event Processor компонентів.

Далі розглянемо останній рівень — пошук і аналіз даних (Data Searches).

- **Пошук даних (Data Searches)**

На завершальному етапі нормалізовані та оброблені дані передаються до останнього рівня в QRadar, де вони стають доступними для пошуку користувачами. Це дозволяє аналізувати, створювати звіти та розслідувати сповіщення, правила та інциденти через консоль QRadar. Консоль також надає можливість аналітикам виконувати дії та адміністративні завдання.

У розподілених середовищах консоль QRadar не виконує обробку подій і потоків, а також не зберігає дані. Натомість вона використовується як інтерфейс користувача, де можна здійснювати пошуки, переглядати звіти, отримувати сповіщення та проводити розслідування.

- **Приклади архітектури.**
  - ✓ **Розподілена архітектура.** Приклад 1: Кілька колекторів і процесорів, підключених до консолі (рис.7.09), допомагають розподілити навантаження між декількома компонентами та забезпечують розділення середовищ.

✓ **Розподілена архітектура.** Приклад 2: У цьому випадку маємо 2 Data Collection з 1 процесором подій (рис.7.09)

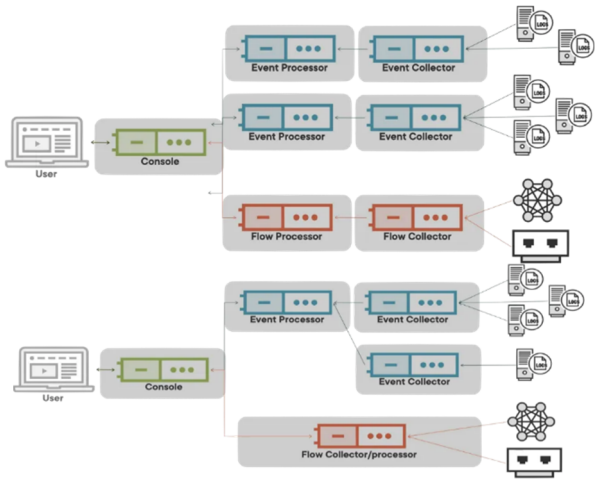


Рис. 7.09. Приклади розподіленої архітектури QRadar

✓ **Концепція централізованої архітектури QRadar** передбачає, що всі його компоненти працюють в межах єдиного серверного рішення (рис.7.10). Такий підхід найчастіше використовується в малих та середніх компаніях, де обсяг подій і мережного трафіку невеликий. QRadar має спеціальний компонент "все в одному" (All-in-One), який об'єднує всі ключові функції системи: збір даних, обробку подій та потоків, а також забезпечує інтерфейс користувача. Це дозволяє розгорнути SIEM-рішення без необхідності окремих серверів для різних функцій.

✓ **Мультитенантність (мультиорендність) у QRadar** означає можливість обслуговування кількох клієнтів в одному спільному середовищі. Замість того щоб розгортати окреме середовище для кожного клієнта, можна використовувати єдину консоль, яка керує декількома процесорами подій і потоків, а також колекторами. При цьому дані кожного клієнта залишаються ізольованими, що гарантує безпеку та конфіденційність інформації.

✓ **Хмарне розгортання** - QRadar може працювати як локальне рішення (on-premises) (рис.7.10) або як хмарне рішення (QRadar on Cloud, QROC), що дозволяє організаціям обирати зручний варіант.

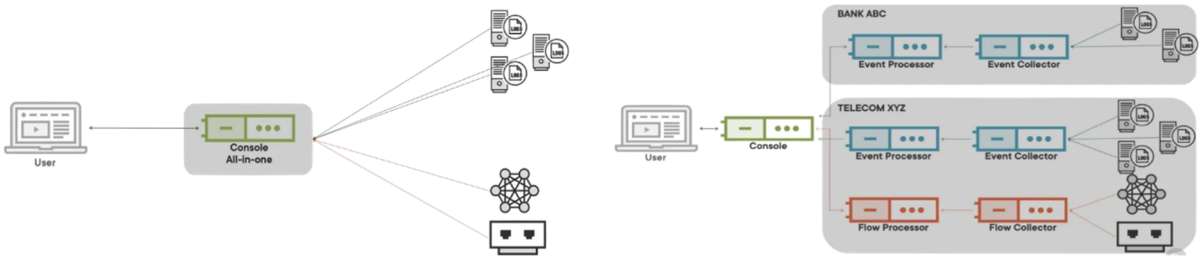


Рис. 7.10. Приклади централізованої та мультитенантної архітектури QRadar

Ліцензування IBM QRadar

QRadar має гнучку систему ліцензування, яка залежить від конкретного модуля та обсягу оброблюваних подій. Основні підходи:

- ✓ Ліцензія на обсяг подій (EPS - Events Per Second). Використовується для SIEM і визначає кількість подій, які система може обробити за секунду. Наприклад, організація може придбати ліцензію на 5000 EPS або більше.
- ✓ Ліцензія на мережний трафік (FPM - Flows Per Minute). Використовується для аналізу мережного трафіку (QNI, Network Insights). Дозволяє обробляти певну кількість мережних потоків на хвилину.
- ✓ Ліцензія для QRadar on Cloud (QROC). Передбачає підписку на SaaS-рішення з оплатою за використання ресурсів.
- ✓ Модульні ліцензії

Цікаві факти про IBM QRadar

- ✓ IBM QRadar входить до списку лідерів ринку SIEM-рішень за оцінками Gartner та Forrester.
- ✓ Включає вбудований механізм машинного навчання для виявлення складних атак.
- ✓ Поєднує SIEM та SOAR в єдиній платформі, що спрощує реагування на інциденти.
- ✓ Має власний маркетплейс (IBM Security App Exchange) для розширення функціоналу.

5. Огляд Elastic Security.

Elastic Security - окремий продукт, створений на базі ELK Stack, який спеціалізується на кібербезпеці. У цій темі доречніше розглянути саме його, а не більш відомий та розпіарений ELK Stack - універсальний набір інструментів для роботи з логами та даними. Elastic Security – це спеціалізоване рішення для кібербезпеки, яке базується на ELK Stack. Тобто Elastic Security є частиною екосистеми ELK, але з додатковими можливостями для безпеки [14].

- ➢ **ELK Stack (Elastic Stack)** - набір інструментів для збору, зберігання, аналізу та візуалізації логів і даних. Він складається з:
  - ✓ **Elasticsearch** – пошуковий та аналітичний рушій для зберігання та обробки даних.
  - ✓ **Logstash** – інструмент збору, обробки та перенаправлення логів.
  - ✓ **Kibana** – інтерфейс для візуалізації даних та аналітики.
  - ✓ **Beats** – легкі агенти для збору даних із різних джерел.

Elastic Stack використовується для моніторингу логів, бізнес-аналітики, аналізу продуктивності та багатьох інших задач.

- **Elastic Security**, створений на базі ELK Stack, який спеціалізується на кібербезпеці. Він включає:
  - ✓ Захист кінцевих точок (**Endpoint Security**) – запобігання та виявлення загроз на пристроях.
  - ✓ **SIEM (Security Information and Event Management**) – збір та аналіз подій безпеки в реальному часі.
  - ✓ **Threat Hunting** – активний пошук загроз у даних.
  - ✓ Аналітику загроз (**Threat Intelligence**) – інтеграцію зі сторонніми джерелами загроз.

Elastic Security поєднує функції виявлення загроз у SIEM із засобами запобігання та реагування на інциденти на кінцевих точках в єдиному рішенні. Ці аналітичні та захисні можливості, посилені швидкістю та розширюваністю Elasticsearch, дозволяють аналітикам захищати організацію від загроз ще до того, як вони спричинять шкоду чи втрати.

Ключові можливості та переваги Elastic Security:

- ✓ **Централізоване управління безпекою**: єдиний інтерфейс для моніторингу, виявлення та реагування на загрози в масштабі всієї інфраструктури.
- ✓ **Detection Engine**: ефективне виявлення атак, помилкових конфігурацій та підозрілої активності на основі правил і попередньо налаштованих задач машинного навчання, включно з безсигнатурними атаками.
- ✓ **Інтерактивні візуалізації**: потужні засоби для дослідження подій, аналізу зв'язків між процесами та розслідування інцидентів.
- ✓ Вбудоване управління інцидентами (**Case Management**): можливість створювати, відслідковувати інциденти та автоматизувати дії у відповідь.
- ✓ **Інтеграція з популярними платформами**: гнучка взаємодія з зовнішніми сервісами для спрощення обробки інцидентів та розширення функціональності.
- ✓ **Потужна автоматизація та гнучке адміністрування**: можливість налаштовувати, масштабувати та інтегрувати рішення під потреби організації.

Компоненти та робочий процес Elastic Security

Діаграма на рис.7.11 наочно ілюструє робочий процес Elastic Security. Короткий огляд процесу та його основних компонентів. Дані надходять з різних джерел, в тому числі з хостів, до **Elasticsearch** у такий спосіб:

- **Elastic Endpoint Security** – інтеграція Elastic Agent, яка захищає хости від шкідливого програмного забезпечення та передає наступні набори даних:
  - ✓ Windows: процеси, мережні з'єднання, файли, DNS, реєстр, завантаження DLL та драйверів, виявлення шкідливого ПЗ
  - ✓ Linux/macOS: процеси, мережні з'єднання, файли
- Інтеграції (**Integrations**) – зручний спосіб надсилання даних у Elastic Stack. Доступні інтеграції для популярних сервісів та платформ, таких як Nginx, AWS, MongoDB, а також для універсальних джерел, як-от лог-файли.
- Модулі Beat (**Beat modules**) – легковагові агенти збору даних. Вони дозволяють отримувати та обробляти специфічні набори даних із поширених джерел, таких як події хмарних сервісів, журнали ОС та метрики. До основних модулів, пов'язаних із безпекою, належать:

Отримані дані уніфікуються відповідно до **Elastic Common Schema (ECS)** і зберігаються у **Elasticsearch** для подальшого аналізу.

Додаток **Elastic Security** у **Kibana** використовується для керування Detection engine, Cases та Timeline, а також для адміністрування хостів із запущеним Endpoint Security.

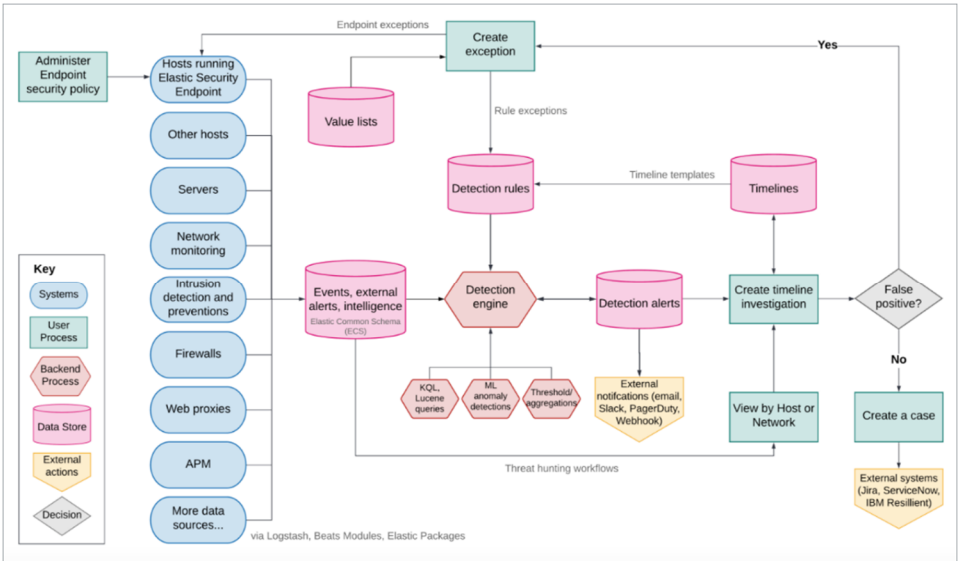


Рис. 7.11. Компоненти та робочий процес Elastic Security

- **Detection engine** – автоматичний пошук підозрілої активності хостів і мережі за допомогою:
  - ✓ Правил виявлення (**Detection rules**) – періодично аналізують дані (індекси Elasticsearch), отримані з ваших хостів, на предмет підозрілих подій. Якщо така подія виявлена, створюється сповіщення. Сповіщення можуть надсилатися у Slack, електронну пошту чи інші зовнішні системи. Ви можете створювати власні правила або використовувати попередньо налаштовані.

- ✓ Винятків (**Exceptions**) – дозволяють зменшити шум і кількість хибнопозитивних спрацьовувань. Винятки прив'язані до правил і блокують створення сповіщень, якщо виконуються певні умови. Списки значень (Value lists) містять дані про події, які можуть бути використані як критерії винятків. Якщо Elastic Endpoint Security встановлено на ваших хостах, можна додавати винятки для шкідливого ПЗ безпосередньо у додатку Security.
- ✓ Задач машинного навчання (**Machine learning jobs**) – автоматичне виявлення аномалій у подіях хостів і мережі. Для кожного хоста визначаються бали аномальності (Anomaly scores), які можна використовувати в правилах виявлення загроз.
- ✓ **KQL/Lucene queries, Thresholds/Aggregations** – для кастомного пошуку та агрегацій.

Після спрацьовування правил

- **Detection Alerts** – тригери можуть бути надіслані в Slack, Email, Jira, ServiceNow, Webhooks тощо.
- **Timeline** – робочий простір для розслідування подій та сповіщень. Використовує запити (queries) та фільтри (filters) для деталізації подій, пов'язаних із конкретним інцидентом. Шаблони Timeline (Timeline templates) можуть бути прикріплені до правил та використовуватися при розслідуванні сповіщень.Збереження та обмін Timeline – дозволяє ділитися розслідуваннями з іншими користувачами та додавати їх до Cases
- **Cases** – внутрішня система управління інцидентами безпеки. Дозволяє створювати, відстежувати та обмінюватися інформацією про інциденти безпосередньо у Security. Інтегрується із зовнішніми системами управління заявками
- **Адміністрування.** Перегляд та керування хостами, на яких працює Elastic Endpoint Security. Надсилання даних у Elastic Security. Процес збору даних для Elastic Security та встановлення інтеграції Endpoint Security описано у відповідних розділах документації.

Ліцензування. Elastic Security доступний як у безкоштовній, так і у платних версіях.

1. Безкоштовна версія (**Basic License**) Elastic Security включена у безкоштовну версію Elastic Stack. Базові функції:
   - ✓ Збір, пошук та аналіз логів
   - ✓ Візуалізація даних у Kibana
   - ✓ Фільтрація, обробка та збагачення подій безпеки
   - ✓ Ручне дослідження загроз та аналіз інцидентів

   доступні без додаткових витрат.

2. Платні плани (**Gold, Platinum, Enterprise License**). Щоб отримати розширені можливості, потрібна комерційна ліцензія (Elastic Security for Elasticsearch Service). У платних версіях додається:
   - ✓ Автоматизоване виявлення загроз (Detection Engine) – правила кореляції та аналітика загроз.
   - ✓ Автоматизація реагування (SOAR-функції) – інтеграція з SIEM/SOAR рішеннями, керування інцидентами.
   - ✓ Endpoint Security – розширений захист кінцевих точок із можливістю блокування атак.
   - ✓ Машинне навчання для поведінкового аналізу – виявлення аномалій.
   - ✓ Доступ до підтримки Elastic – технічна допомога та консультації.

Безкоштовної версії достатньо для базового моніторингу та аналізу загроз, але для повноцінного SIEM-рішення з автоматизацією, інтеграціями та захистом кінцевих точок рекомендується платна ліцензія. Детальніше про актуальні умови ліцензування можна подивитися на офіційному сайті Elastic.

6. Огляд MS SCOM як інструменту системного моніторингу

Microsoft SCOM (System Center Operations Manager) відстежує різні комп'ютери, пристрої, програми та служби в корпоративному IT-середовищі та повідомляє адміністраторам, які з цих об'єктів моніторингу справні, а які – ні. Він також надсилає сповіщення, коли виявляє проблеми, надає корисну інформацію про виявлену проблему, знаходить її причину та впроваджує можливе рішення [32].

Для виконання всіх вищезазначених завдань різні компоненти працюють разом у SCOM. Ці компоненти є частиною групи керування, яка створюється під час інсталяції SCOM і є основною функціональною одиницею. Компоненти можуть існувати на одному сервері або можуть бути розподілені між кількома серверами. Компоненти SCOM включають наступне:

- **Оперативна база даних**. База даних SQL Server , яка містить усі конфігураційні дані та зберігає дані моніторингу на короткий термін.
- **База даних сховища даних.** Також база даних SQL Server, хоч і така, яка зберігає дані моніторингу та сповіщень для історичних цілей, тобто довгострокового зберігання.
- **Сервер керування.** Адмініструє групу керування та спілкується з базою даних.
- **Сервер звітності.** Створює та представляє звіти, створені з даних у базі даних сховища даних .

- **Агент.** Служба, встановлена на комп'ютері для збору даних моніторингу, створення сповіщень і запуску відповідей; він повідомляє серверу керування в групі керування.
- **Послуга Агент моніторингу.** Збирає дані про продуктивність, виконує завдання та надсилає зібрані дані на сервер керування.

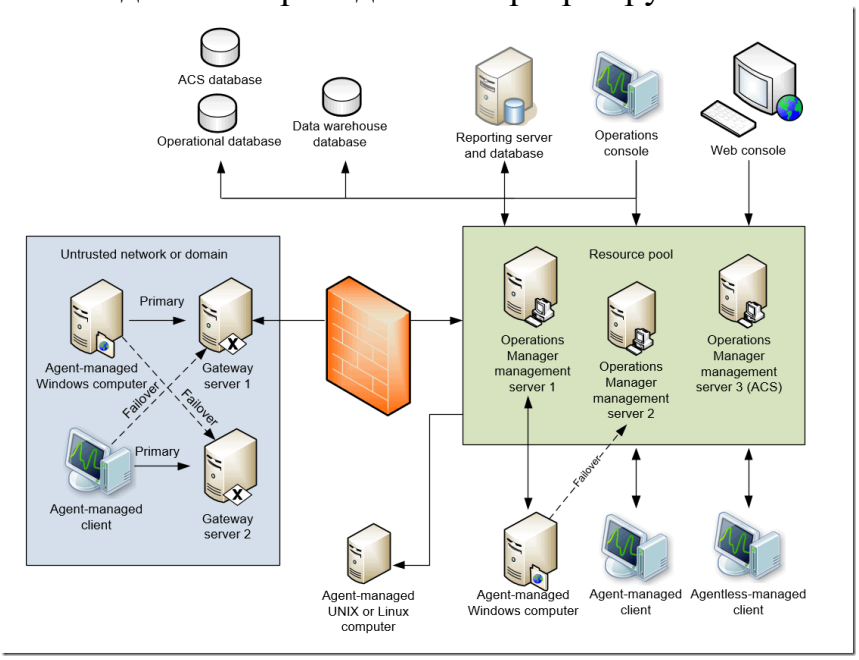


Рис. 7.12 Архітектура SCOM

SCOM, як вказує його назва, не є вузькоспеціалізованим інструментом моніторингу MS. Це скоріше набір програмного забезпечення для розгортання, налаштування, керування та моніторингу всіх серверів, компонентів і служб IT-інфраструктури Windows.

Operations Manager призначений для моніторингу всієї інфраструктури, тому часто це складний звір із багатьма рухомими частинами, попередніми умовами та залежностями, ролями сервера, агентами тощо. Корпорація Майкрософт детально описує їх у своєму посібнику щодо системних вимог для SCOM, але ми трошки пізніше коротко пробіжимося по цим вимогам. Архітектура SCOM виглядає приблизно так, як показано на рис. 7.12.

Агенти Microsoft SCOM

У System Center Operations Manager агент — це служба, встановлена на комп'ютері, яка шукає дані конфігурації і заздалегідь збирає відомості для аналізу та створення звітів, вимірює стан працездатності об'єктів, що відстежуються, таких як база даних SQL або логічний диск, і виконує завдання на вимогу оператора або у відповідь умову. Вона дозволяє Operations Manager відстежувати операційні системи Windows, Linux та UNIX, а також компоненти IT-служби, встановлені на них, наприклад веб-сайт або контролер домену Active Directory.

- **Агент Windows**

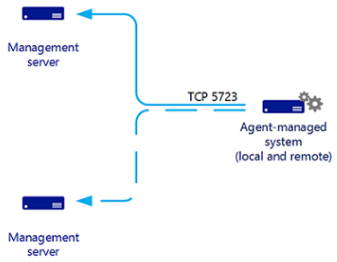
На відстежуваному комп'ютері Windows агент Operations Manager вказаний як служба Microsoft Monitoring Agent (MMA). Служба Microsoft Monitoring Agent збирає дані про події та продуктивність, виконує завдання та інші робочі процеси, визначені в пакеті керування. Навіть якщо ця служба не може підключитися до сервера керування, якому вона підпорядковується, вона продовжує працювати і поміщає зібрані дані та події в чергу на диску комп'ютера, що спостерігається. Під час відновлення підключення служба Microsoft Monitoring Agent надсилає зібрані дані та події на сервер керування. Іноді Microsoft Monitoring Agent називають службою працездатності.

Служба Microsoft Monitoring Agent також працює на серверах керування. На сервері керування ця служба виконує робочі процеси моніторингу та керує обліковими даними. Для запуску робочих процесів служба викликає процеси MonitoringHost.exe за допомогою вказаних облікових даних. Ці процеси виконують спостереження та збирають дані журналів подій, дані інструментарію керування Windows (WMI), а також виконують такі дії, як запуск скриптів. Агент Operations Manager надсилає попередження та дані виявлення на призначений основний сервер управління, який записує ці дані до робочої бази даних. Крім того, агент відправляє дані про події, продуктивність та стан на основний сервер управління, який одночасно записує ці дані в робочу базу даних і в базу даних сховища даних.

Агент надсилає дані відповідно до параметрів розкладу для кожного правила та монітора. У разі оптимізованих правил збору даних дані передаються тільки в тому випадку, якщо вибірка лічильника відрізняється від попередньої вибірки на вказану величину допуску, наприклад, на 10%. Це допомагає скоротити мережний трафік та обсяг даних, що зберігаються у робочій базі даних.

Крім того, всі агенти регулярно відправляють пакет даних, званий пульсом , на сервер управління: за умовчанням це відбувається кожні 60 секунд. Мета пульсу полягає у перевірці доступності агенту та зв'язку між агентом та сервером управління. Детальний опис реалізації механізму пульс див. у статті How Heartbeats Work in Operations Manager (Принципи роботи пульсу в Operations Manager) .

Якщо коротко, то SCOM використовує тактові сигнали для моніторингу каналів зв’язку між агентом і основним сервером керування агентом. Пакет даних регулярно, за замовчуванням кожні 60 секунд, надсилається від агента



на сервер керування через порт 5723 (TCP) (рис.7.13).

Рис. 7.13 Management server - Agent-managed host

Якщо агент чотири рази не може надіслати Heartbeats сигнал, створюється сповіщення про збій серцевого ритму служби працездатності, і сервер керування намагається зв'язатися з комп'ютером за допомогою ping. Якщо комп'ютер не відповідає на запит ping, створюється сповіщення, що не вдалося підключитися до комп'ютера. Для кожного агенту, Operations Manager запускає спостерігач служби працездатності, який спостерігає стан віддаленої служби працездатності з точки зору сервера управління. Агент взаємодіє із сервером управління через TCP-порт 5723.

- **Агент Linux/UNIX**

Архітектура агенту UNIX та Linux істотно відрізняється від архітектури агенту Windows. Агент Windows має службу працездатності, відповідальну за оцінку працездатності комп'ютера, що відстежується. Агент UNIX та Linux не запускає службу працездатності. Натомість він передає відомості до служби працездатності на сервері управління для оцінки. На сервері керування запускаються всі робочі процеси для моніторингу стану операційної системи, визначені у реалізації пакетів керування UNIX та Linux:

- ✓ Диск
- ✓ Процесор
- ✓ Пам'ять
- ✓ Мережні адаптери
- ✓ Операційна система
- ✓ Процеси
- ✓ Файли журналу

Агенти UNIX та Linux для Operations Manager складаються з диспетчера об'єктів CIM (тобто сервера CIM) та набору постачальників CIM.

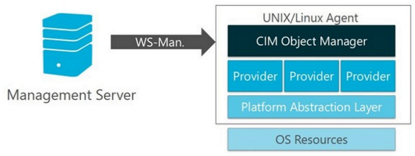


Рис. 7.14 Management server – Unix/Linux host

CIM - Common Information Model – модель даних, яка використовується для представлен-ня та обміну інформацією про керовані об'єкти в середовищі SCOM. Диспетчер об'єктів CIM - це серверний компонент, що реалізує WS-Management зв'язку, автентифікації, авторизації та відправки запитів постачальникам (рис.7.14).

WS-Management (Web Services Management) – відкритий стандарт, який описує протокол на основі SOAP для управління серверами, пристроями, програмами та різними веб-сервісами. Постачальники є ключовим елементом реалізації CIM в агенті, визначаючи класи та властивості CIM, взаємодіючи з API ядра для отримання необроблених даних, форматуючи

дані (наприклад, обчислюючи різниці та середні значення) та обслуговуючи запити, надіслані диспетчером об'єктів CIM. В операційних системах з System Center Operations Manager 2007 R2 по System Center 2012 SP1 диспетчер об'єктів CIM, що використовується в агентах UNIX та Linux Operations Manager, є сервером OpenPegasus. Постачальники, які використовуються для збору даних моніторингу та складання відповідних звітів, розробляються Майкрософт та надаються на сайті CodePlex.com з відкритим кодом.

Обмін даними між сервером управління та агентом UNIX та Linux ділиться на дві категорії: обслуговування агента та моніторинг працездатності. На сервері керування для взаємодії з комп'ютером UNIX або Linux використовуються два протоколи:

- ✓ Secure Shell (SSH) та протокол SFTP. Використовується для завдань з обслуговування агента, включаючи встановлення, оновлення та видалення агентів.
- ✓ Веб-служби для керування (WS-Management). Використовується для всіх операцій моніторингу та виявлення вже встановлених агентів.

Стандартна інсталяція SCOM — це, по суті, лише базова структура, на якій можна розмістити пакети керування. Кожен пакет визначає як структуру програми або служби, яку слід відстежувати, так і всі її компоненти та їхні взаємозв’язки (модель служби), а також дані, які слід збирати для оцінки працездатності цієї програми (модель працездатності).

Агент автоматично виявляє всі попередньо визначені «об’єкти» в моделі обслуговування для пакета керування. Наприклад, для SQL Server це включатиме бази даних, файли баз даних, завдання тощо. Для кожного об’єкта (логічно згрупованого в класи) він збирає дані моніторингу, визначені моделлю працездатності, і надсилає їх назад на сервер керування, який зберігає дані моніторингу та сповіщень в операційній базі даних (тут зберігається поточна звітність) і в сховищі даних (для історичних даних).

SCOM — це гарний інструмент моніторингу на системному рівні для звітування про загальний стан і продуктивність різноманітної серверної інфраструктури та служб, які на ній працюють. Він збирає дані про широкий спектр системних показників, служб, станів процесів і лічильників продуктивності Windows для кожного сервера. Він відстежуватиме журнали подій сервера. Він повідомлятиме про будь-які невдалі входи на сервер та інші помилки та попередження сервера. Він попередить вас про проблеми з процесором, пам’яттю чи вводом/виводом або помилки мережі. Він сповістить вас, коли на фізичному диску чи логічному томі буде мало місця для зберігання.

Проблеми справді виникають, лише якщо ви намагаєтеся розширити функціональність SCOM для моніторингу складних програм або служб, таких як SQL Server. SCOM не розроблявся для того, щоб заглиблюватися в «нутрощі» складної служби, щоб допомогти зрозуміти, що саме там

відбувається, коли щось йде не так. Продукт створений та задуманий як спосіб надання загальної картини моніторингу.

Налаштування правил, моніторів і сповіщень SCOM для пакетів керування певними конкретними ролями є складною справою, що ускладнюється відносно поганою документацією. Для цього потрібні спеціальні знання як SCOM, так структури та алгоритмів ролі чи служби, моніторинг якої налаштовується. Зазвичай, налаштування такого моніторингу за допомогою SCOM передбачає написання користувацьких правил і моніторів, або, в крайньому випадку, навіть написання власного пакету керування. Навіть після того, як усе налаштовано та запрацює, багато команд все одно виявляють, що отримані дані моніторингу важко зрозуміти. Знову ж таки, для інтерпретації даних і дій на основі них потрібна спеціальна експертиза. Складність для організації такого моніторингу полягає в розумінні межі між тим, що SCOM може і повинен контролювати, і тим, що можна зробити краще та більш масштабованим способом за допомогою інструменту, розробленого спеціально для моніторингу спеціалізованих служб.

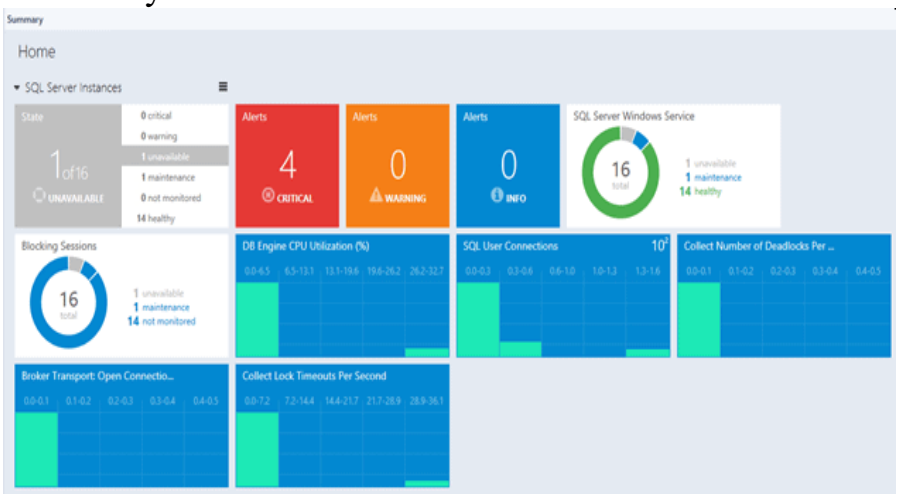


Рис. 7.15 Інформаційна панель SCOM

Таким чином, SCOM є чудовим інструментом системного рівня, але в ньому будуть відсутні важливі діагностичні дані та дані для усунення несправностей, необхідні для дослідження деталей проблеми продуктивності нижчого рівня, таких як статистика виконання, плани запитів, стани очікування, ланцюг блокування тощо.

Щоб зовсім Вас не розчарувати у можливостях SCOM розглянемо його інформаційну панель (рис.7.15). Користувач сам налаштовуєте інформаційну панель із інформацією, яку хоче бачити. Якщо необхідно побачити базову лінію для метрики, щоб порівняти поточну поведінку з «нормою», потрібно буде створити базову лінію. Як зазначалося раніше, команди часто створюють необхідну інформаційну панель фрагментарно, оскільки вони дізнаються, які дані їм дійсно потрібні для діагностики

кожного типу проблеми. Це - класичний, нормальний підхід до побудови гарної системи моніторингу.

Загальним результатом є те, що, хоча SCOM пропонує єдиний координаційний центр для всього моніторингу всіх систем, він недоступний для тих у IT-команді, яким може знадобитися реагувати на попередження, але яким бракує повних, детальних знань про домен всі елементи та складові інфраструктури, що моніториться. Але, погодьтеся, такі знання необхідні для побудови будь якої моніторингової системи.

Системні вимоги для SCOM

Щоб отримати максимальну користь від SCOM, важливо спочатку перевірити системні вимоги та переконатися, що ці вимоги виконуються. Тим не менш, SCOM розроблений як гнучкий і масштабований, тому вимоги до апаратного та програмного забезпечення для певних сценаріїв можуть відрізнятися від наведених нижче рекомендацій.

- Необхідно дотримуватися рекомендованих обмежень для всіх контрольованих елементів, включаючи комп'ютери, які контролює агент, консолі одночасних операцій, комп'ютери, керовані агентом і Unix або Linux на групу керування, а також мережні пристрої, якими керує пул ресурсів із трьома або більше серверами керування.
- Кількість додатків для моніторингу продуктивності додатків має бути менше 400.
- Кількість URL-адрес, які відстежуються на агента, має бути менше 50.
- Мінімум 8 гігабайт пам'яті та 10 ГБ дискового простору потрібні для конфігурації таких ролей, як сервер керування, сервер шлюзу, який керує до 2000 агентів, сервер веб-консолі та сервер SQL Server Reporting Services.
- Для будь-якої ролі сервера SCOM мінімальною вимогою до процесора x64 є чотириядерний центральний процесор 2,66 ГГц.
- Для налаштування сервера керування Operations Manager потрібна мінімальна версія Windows Server 2019 Standard або Datacenter.
- Для компонента сервера звітів Operations Manager потрібна версія Windows Server 2019 або Windows Server 2022 Standard або Datacenter.

Серед інших мінімальних системних вимог для налаштування Microsoft SCOM:

- Internet Explorer (IE) 11 і Silverlight 5 для забезпечення зворотної сумісності клієнтських браузерів із інформаційними панелями з підтримкою Silverlight.
- Windows PowerShell версії 2.0 або 3.0 для консолі Operations Manager. Налаштування Command Shell Operations Manager виконується у консолі SCOM розділ "Administration" (Адміністрування), вкладці "Settings" (Налаштування) - "Security" (Безпека) активацією відповідної опції.
- .NET Framework 4.7.2 або 4.8 для сервера керування та сервера шлюзу.

- Microsoft Edge версії 88, IE версії 11, Google Chrome версії 88 для клієнтського веб-браузера для веб-консолі HTML5.
- Налаштований протокол передачі гіпертексту або безпечне прив’язування HTTP .
- Клієнтська операційна система Windows 10 і Windows 11 .
- Служби доменів Active Directory (AD DS) справні та підтримуються на певних мінімальних рівнях конфігурації. AD DS використовується для керування автентифікацією та авторизацією користувачів, а також для розподілу та організації об'єктів, таких як сервери, комп'ютери, служби та ролі, у доменній мережі. Мінімальні рівні конфігурації AD DS зазвичай відповідають певним версіям операційної системи Windows Server, які підтримуються SCOM. Наприклад, у хмарній реалізації, використовується Azure Active Directory (Azure AD) від Microsoft, можливе також використання гібридної служби доменів. Основні аспекти, пов'язані з використанням дерева каталогів у роботі SCOM, включають:
  - ✓ **Автентифікація і авторизація**. SCOM використовує службу доменів AD DS для автентифікації користувачів та надання їм відповідних дозволів на доступ до ресурсів системи моніторингу.
  - ✓ **Розподіл об'єктів**. Об'єкти, що моніторяться SCOM, такі як сервери, комп'ютери, програмне забезпечення та служби, організовуються відповідно до структури дерева каталогів AD DS. Це дозволяє керувати об'єктами моніторингу та надавати доступ до них з різних компонентів SCOM.
  - ✓ **Ролі та дозволи**. AD DS дозволяє налаштовувати ролі та дозволи користувачів SCOM з використанням Role-Based Access Control (RBAC). Це дозволяє диференціювати доступ користувачів до функціональності SCOM відповідно до їхніх обов'язків та повноважень.
  - ✓ **Розгортання і реплікація**. AD DS може бути використане для керування процесом розгортання та реплікації SCOM, дозволяючи ефективно розподіляти об'єкти моніторингу та забезпечувати доступність служби моніторингу у різних географічних регіонах.
- Встановлена та справна для належної підтримки AD DS і SCOM система доменних імен.

Висновки до розділу

Ефективна побудова систем моніторингу та кібербезпеки потребує комплексного, стратегічно виваженого підходу. Жоден окремий інструмент, навіть настільки потужний, як Elastic Security, не здатен повністю задовольнити всі вимоги сучасного підприємства або організації. Справжня ефективність досягається шляхом інтеграції та грамотного поєднання

різноманітних рішень: як уже розглянутих нами Elastic Stack компонентів, так і інших популярних систем моніторингу, таких як Nagios, Zabbix, Grafana, Prometheus, а також засобів SIEM, IDS/IPS та спеціалізованих аналітичних платформ.

Кожен з продуктів має свої сильні сторони, можливості та обмеження. Наприклад, одні інструменти краще підходять для збору й візуалізації метрик, інші — для обробки логів, кореляції подій, або виявлення аномалій у мережному трафіку. Важливо враховувати специфіку інфраструктури, завдання організації, рівень критичності окремих систем та доступні ресурси для обслуговування обраних рішень.

Таким чином, комплексна система моніторингу та безпеки повинна бути побудована як єдиний цілісний механізм, у якому різні компоненти доповнюють один одного, забезпечуючи багаторівневу видимість, раннє виявлення загроз та своєчасне реагування. Лише поєднання різноманітних технологій та їх правильна інтеграція дозволяють досягти високого рівня надійності та стійкості інформаційної системи в умовах сучасних кіберзагроз.

Контрольні запитання

1) Які етапи включає виявлення порушень безпеки за допомогою Wazuh FIM?
2) Які шаблони індексів використовує Wazuh Indexer?
3) Які основні серверні компоненти Wazuh і їхні функції?
4) Як у SCOM працюють тактові сигнали (Heartbeats) між агентом і сервером?
5) З яких компонентів складається архітектура SCOM?
6) Як активується інтерфейс командного рядка SCOM Command Shell?
7) Які бази даних входять у типову архітектуру SCOM?
8) Які вимоги необхідні для налаштування серверів керування та шлюзу Operations Manager?
9) Яку роль виконує Snort?
10) Що являє собою Splunk Enterprise Security?
11) Які інформаційні панелі включає Wazuh Dashboard?
12) Де можна перевірити інформацію про індекси індексатора Wazuh?
13) Які можливості забезпечує Wazuh?
14) У чому полягає ключова можливість Splunk ES?
15) У якому режимі Snort не здійснює блокування трафіку?
16) Який компонент Wazuh відповідає за візуалізацію даних?
17) Які основні три рівні архітектури IBM QRadar?

18) Які типи даних збирає QRadar і як вони використовуються?
19) У чому полягає роль модуля Custom Rules Engine (CRE) в архітектурі QRadar?
20) Які основні підходи до ліцензування IBM QRadar застосовуються?

Розділ 8. Системи моніторингу та стратегії реагування на інциденти кібербезпеки

1. Застосування моніторингу для виявлення загроз безпеці.

У сучасному світі кібербезпека є однією з найважливіших проблем для будь-якої організації. Кіберзлочинці постійно вдосконалюють свої методи, і традиційних методів захисту вже недостатньо [31]. Моніторинг - це потужний інструмент, який може допомогти:

- **Виявити загрози на ранній стадії**, перш ніж вони завдадуть шкоди інформаційній системі.
- **Швидко відреагувати на інциденти**, швидко визначити масштаби інциденту та вжити заходів для його усунення.
- **Запобігти майбутнім атакам** шляхом виявлення вразливостей у інформаційній системі та вжити заходів для їх усунення.

Коли мова заходить про застосування моніторингу для виявлення загроз безпеці, мови про розділення типів моніторингу не може бути, бо для цієї благородної цілі застосовуються всі допустимі типи моніторингу, крім заборонених законом ☺

Моніторинг – це, по суті, процес отримання інформації про події, які вже відбулися. Таким чином, коли ми говоримо про застосування моніторингу для виявлення загроз безпеці потрібно розуміти, що доводити можливі загрози безпеці до здійснення не потрібно. Потрібно передбачити за результатами моніторингу ці загрози чи навіть їх можливість.

> Моніторинг безпеки – це практика збору інформації на різних рівнях робочого навантаження (інфраструктура, програми, операції) для отримання інформації про підозрілу діяльність . Мета полягає в тому, щоб передбачити інциденти та вивчити минулі події. Дані моніторингу є основою для аналізу того, що сталося після інциденту, щоб допомогти реагувати на інциденти та провести розслідування.

- **Ключові стратегії дизайну моніторингу безпеки.**

  Основною метою моніторингу безпеки є виявлення загроз . Необхідно запобігти потенційним порушенням безпеки та підтримувати безпечне середовище. Однак не менш важливо визнати, що не всі загрози можливо запобіжно заблокувати. У таких випадках моніторинг також слугує механізмом для визначення причини інциденту безпеки, який стався, незважаючи на зусилля із запобігання.

  До моніторингу необхідно підходити з різних точок зору:

- ✓ **Моніторинг на різних рівнях.** Спостереження з різних рівнів — це процес отримання інформації про
  - потоки користувачів: як користувачі взаємодіють з додатком, веб-сайтом, хостом. Може включати дані про те, які сторінки вони відвідують, на які кнопки натискають, скільки часу вони проводять на кожній сторінці тощо.
  - доступ до даних: отримання доступу до даних. Може включати дані про те,
    - які користувачі отримують доступ до яких даних,
    - коли вони отримують доступ до даних і
    - як вони отримують доступ до даних (наприклад, через веб-інтерфейс або API).
  - ідентичність: співставлення користувача з конкретною людиною або службово-функціональним обліковим записом. Може включати дані про те,
    - як користувачі аутентифікуються у системі,
    - які у них ролі та дозволи, а також
    - їхню контактну інформацію.
  - мережу: як працює мережа. Може включати дані про те,
    - які пристрої підключені до мережі,
    - який трафік проходить через мережу та
    - як мережа використовується.
  - операційну систему: робота операційних систем всіх рівнів. Може включати дані про те,
    - які процеси запущені,
    - які ресурси використовуються та
    - які помилки виникають.

  Це лише кілька базових прикладів, що моніториться на різних рівнях. Іншими словами – моніторинг на різних рівнях це - процес збору даних про різні аспекти середовища Кожна зі згаданих областей пропонує унікальну інформацію, яка може допомогти визначити відхилення від очікуваної поведінки, встановленої порівняно з базовим рівнем безпеки. І навпаки, безперервний моніторинг системи та додатків протягом тривалого часу може допомогти встановити цю базову позицію. Наприклад, спостерігається близько 1000 спроб входу у систему ідентифікації щогодини. Якщо Ваш моніторинг виявляє сплеск 50 000 спроб входу протягом короткого періоду, можливо, зловмисник намагається отримати доступ до системи і цей сплеск активності - прояви спроби злому.

- ✓ **Моніторинг при різних масштабах впливу.** Припустімо, що користувач випадково отримав підвищені привілеї, або сталося

порушення безпеки. Якщо користувач виконує дії, що виходять за межі його визначеної сфери, вплив може обмежуватися діями, які можуть виконувати інші користувачі.

Однак, якщо будь яка внутрішня сутність (користувач чи криво написаний тригер) скомпрометує базу даних, ступінь потенційної шкоди залишається невизначеним. Якщо компрометація виникає на стороні ресурсу ядра системи (наприклад MS AD чи AWS AD), вплив може бути глобальним і розповсюдиться на всі сутності, які взаємодіють із ресурсом. Радіус заподіяної компроментацією шкоди, або масштаб удару можуть значно відрізнятися залежно від того, який із сценаріїв має місце. Додаткові поради:

- Використовуйте принцип найменших привілеїв, щоб надати користувачам лише ті права, які їм дійсно потрібні.
- Регулярно проводьте аудит безпеки для виявлення та усунення вразливостей.
- Впровадьте систему раннього виявлення та реагування на інциденти (EDR).

Визначення різних масштабів впливу:

- ❖ **Локальний:** Вплив обмежується одним компонентом, модулем або підсистемою. Наприклад - несанкціонований доступ до однієї таблиці в базі даних.
- ❖ **Регіональний:** поширюється на кілька компонентів, модулів або підсистем, але не на всю систему. Приклад: несанкціонований доступ до одного сервера в кластері.
- ❖ **Глобальний:** поширюється на всю систему або на кілька систем. Приклад: несанкціонований доступ до центральної бази даних або доменного контролера.
- ❖ **Катастрофічний:** призводить до повної або часткової втрати працездатності системи, що може мати значні фінансові та репутаційні наслідки. Приклад: шифрування даних зловмисним програмним забезпеченням.
- ❖ **Невизначений:** у даний момент неможливо визначити масштаб впливу. Приклад: початкова фаза розслідування інциденту кібербезпеки.

Важливо зазначити, що масштаб впливу може змінюватися з часом. Наприклад, локальний інцидент може перерости в регіональний або глобальний, якщо його не буде вчасно виявлено та усунено.

- ✓ **Використання спеціалізованих засобів моніторингу.** Дуже важливо налаштувати та підтримувати у «бойовому стані» спеціалізовані

інструменти, що постійно сканують аномальну поведінку, яка може свідчити про атаку. Більшість із цих інструментів мають можливості аналізу загроз, які можуть виконувати прогнозний аналіз на основі великого обсягу даних і відомих загроз. Інструменти моніторингу зазвичай не обмежуються простим збором даних, а також мають глибоке розуміння того, що ці дані означають з точки зору безпеки. Вони можуть виявляти аномалії, аналізувати загрози та навіть робити прогнози на основі цих даних.

Інструменти мають бути інтегровані в платформу або принаймні відповідати платформі, щоб отримувати глибокі сигнали від платформи та робити прогнози з високою точністю. Вони повинні мати можливість своєчасно генерувати сповіщення з достатньою інформацією для проведення належного сортування. І навпаки, використання занадто великої кількості різноманітних інструментів може призвести до ускладнень та перевантаження мережі.

- ✓ **Використання моніторингу для реагування на інциденти.** Зведені та нормалізовані дані, перетворені на оперативну розвідку, дозволяють швидко й ефективно реагувати на інциденти. Моніторинг також допомагає діяти після інциденту. Мета — зібрати достатньо даних, щоб проаналізувати та зрозуміти, що сталося. Процес моніторингу фіксує інформацію про минулі події, щоб покращити можливості реагування та потенційно передбачити майбутні інциденти.

- ➢ **Збирання та зберігання даних, щоб відслідковувати діяльність.**

Мета полягає в тому, щоб підтримувати повний контрольний слід подій, важливих з точки зору безпеки. Логування є найпоширенішим способом фіксації шаблонів доступу. Реєстрацію потрібно виконувати для програми, платформи з прив'язкою до конкретного хосту чи ресурсу.

Для контрольного сліду необхідно встановити, коли та хто пов'язаний з певними діями. Необхідно визначити конкретні терміни виконання дій. Зробіть цю оцінку у своєму моделюванні загроз. Щоб протистояти загрозі відмови, слід створити надійні системи журналювання та аудиту, які призведуть до запису дій і транзакцій.

- ✓ **Потоки користувача програми.** Кожна критична з безпекової точки програма (додаток) має бути розроблена так, щоб забезпечити видимість під час важливих чи критичних подій. Визначте критичні точки у програмі та створіть журнал для цих точок. Наприклад, коли користувач входить у програму, зафіксуйте особу користувача, джерело розташування та іншу відповідну інформацію. Важливо підтверджувати будь-яке підвищення привілеїв користувача, дії, які виконує користувач, і чи отримував користувач доступ до

конфіденційної інформації в безпечному сховищі даних. Відстежуйте дії користувача та сеанс користувача.
Щоб полегшити це відстеження, логування виконується за допомогою структурованого журналювання. Це дає змогу легко стандартизовано виконувати запити та фільтрувати журнали.

Потрібно забезпечити відповідальне ведення журналу, щоб зберегти конфіденційність і цілісність інформаційної системи. Секрети та конфіденційні дані не повинні відображатися в журналах. Пам’ятайте про витік особистих даних та інші вимоги відповідності, коли збираєте дані журналу.

- ✓ **Моніторинг ідентифікації та доступу.** Ведіть ретельний облік шаблонів доступу до програм та змін до ресурсів платформи. Майте надійні журнали активності та механізми виявлення загроз, особливо для дій, пов’язаних із ідентифікацією, оскільки зловмисники часто намагаються маніпулювати ідентифікацією, щоб отримати неавторизований доступ.
  Реалізуйте комплексне журналювання, використовуючи всі доступні точки даних. Наприклад, додайте IP-адресу клієнта, щоб відрізнити звичайну активність користувача від потенційних загроз із неочікуваних місць. Усі події журналювання повинні бути позначені часом з надійного серверу часу, а не часом користувача.
  Фіксуйте всі дії з доступом до ресурсів: хто що і коли робить з ресурсами. Випадки підвищення привілеїв є важливою точкою даних, яку слід реєструвати. Дії, пов'язані зі створенням або видаленням облікового запису, також повинні бути зафіксовані. Ця рекомендація поширюється на паролі, логіни, сертифікати, підписи. Слідкуйте за тим, хто має доступ до секретів і коли вони змінюються. Доступ до паролів, логінів, сертифікатів та підписів користувача має бути лише у цього користувача.
  Хоча протоколювання успішних дій є важливим, записування помилок також необхідне з точки зору безпеки. Документуйте будь-які порушення:
  - користувач, який намагається виконати дію, але зіткнувся з помилкою авторизації
  - спроби доступу до неіснуючих ресурсів
  - будь які інші дії, які здаються підозрілими.

- ✓ **Моніторинг мережі.** Відстежуючи мережні пакети та потоки, їх джерела, призначення та структури, отримуємо видимість шаблонів доступу на рівні мережі. Дизайн сегментації мережі має дозволяти

точки спостереження на межі кожної мережі. Простіше це пояснюється так:

Уявіть, що ваша мережа - це велика дорога.

- Мережні пакети та потоки - це як автомобілі, що їздять по ній.
- Джерела - це місця, звідки виїжджають автомобілі.
- Призначення - це місця, куди їдуть автомобілі.
- Структури - це маршрути, якими їдуть автомобілі.

Моніторинг мережі - це як спостереження за цією дорогою.

Ви хочете знати:

- Хто їздить по цій дорозі (які джерела та призначення)?
- Куди вони їдуть (які маршрути)?
- Як часто вони їздять (які шаблони доступу)?

Для цього ви можете:

- Встановити камери на кордонах (точки спостереження).
- Записувати все, що їде через кордони (реєструвати дані).
- Аналізувати записи, щоб бачити, що відбувається.

Наприклад:

- Ви можете відстежувати підмережі (групи автомобілів)
- Ви можете відстежувати журнали брандмауера (записи про те, кому дозволено або заборонено їздити)

Це допоможе вам:

- Виявити підозрілу активність (наприклад, несанкціонований доступ)
- Вирішити проблеми з мережею (наприклад, затори)
- Забезпечити безпеку вашої мережі (наприклад, запобігти аваріям)

Чим краще ви моніторите свою мережу, тим краще ви можете її захистити.

Існують журнали доступу для вхідних запитів на підключення. У цих журналах записуються IP-адреси джерела, які ініціюють запити, тип запиту (GET, POST) та вся інша інформація, яка є частиною запитів.

Захоплення потоків DNS є важливою вимогою для багатьох організацій. Наприклад, журнали DNS можуть допомогти визначити, який користувач або пристрій ініціював певний запит DNS. Співвідносячи діяльність DNS із журналами автентифікації користувача/пристрою, можливо відстежувати дії окремих клієнтів. Ця відповідальність часто поширюється на робочу групу, особливо якщо вони розгортають щось, що робить запити DNS частиною їхньої роботи. Аналіз трафіку DNS є ключовим аспектом безпеки платформи.

Важливо відстежувати неочікувані DNS-запити або DNS-запити, спрямовані на відомі командні та контрольні кінцеві точки.

> **Компроміс:** реєстрація всіх мережних дій може призвести до великого обсягу даних. Кожен запит із рівня 3 можна записати в журнал потоків, включаючи кожну транзакцію, яка перетинає межу підмережі. На жаль, неможливо зафіксувати лише несприятливі події, оскільки їх можна ідентифікувати лише після їх виникнення.
> Приймайте стратегічні рішення щодо типу подій, які потрібно зафіксувати, і тривалості їх зберігання. Якщо ви не будете обережні, керування даними може виявитися непосильним. Існує також компроміс щодо вартості зберігання цих даних.

Зважаючи на компроміси, слід розглянути, чи вигода від мережного моніторингу робочого навантаження є достатньою для виправдання витрат. Якщо у вас експлуатуються рішення з великим обсягом запитів і ваша система широко використовує керовані хмарні ресурси, вартість може переважити переваги. З іншого боку, якщо у вас є рішення, розроблене для використання віртуальних машин із різними портами та програмами, може бути важливо збирати та аналізувати журнали мережі.

- **Фіксація змін системи.**

Щоб підтримувати цілісність системи, необхідно мати точний і актуальний запис про стан системи. Якщо є зміни, цей запис може бути використаний для оперативного вирішення будь-яких проблем, що виникли.

Процеси сканування мережі також повинні видавати телеметрію. Ключовим є розуміння контексту безпеки подій. Знання того, що ініціювало процес сканування, хто його ініціював і коли він був ініційований, може дати цінну інформацію.

Відстежуйте, коли ресурси створюються та коли вони виводяться з експлуатації. Ця інформація надає цінну інформацію для управління ресурсами та підзвітності.

Також необхідно моніторити переміщення (дрейф) ресурсів. Документуйте будь-які зміни в існуючому ресурсі. Також слідкуйте за змінами, які не завершуються в рамках розгортання на парку ресурсів. Журнали мають фіксувати деталі зміни та точний час, коли вони відбулися.

Отримайте всебічне уявлення, з точки зору виправлення, про те, чи є система актуальною та безпечною. Відстежуйте регулярні процеси оновлення, щоб переконатися, що вони завершуються за планом. Процес апдейту безпеки, який не завершується, слід вважати вразливістю. Також

слід вести інвентаризацію, у якій записуються рівні виправлень та будь-які інші необхідні деталі.

Виявлення змін також стосується операційної системи. Це передбачає відстеження того, додаються чи вимикаються служби. Він також включає моніторинг додавання нових користувачів до системи. Існують інструменти, призначені для націлювання на операційну систему. Вони допомагають у безконтекстному моніторингу в тому сенсі, що вони не націлені на функціональність робочого навантаження. Наприклад, моніторинг цілісності файлів є критично важливим інструментом, який дозволяє відстежувати зміни в системних файлах.

Слід налаштувати сповіщення про ці зміни, особливо якщо ви не очікуєте, що вони відбуватимуться часто.

> Коли ви розгортаєте робочу версію елементу інформаційної мережі, переконайтеся, що сповіщення налаштовано для виявлення аномальної активності. А аномальну активність важливо передбачити у процесі тестування елементу мережі.

Включіть перевірку журналювання та попередження як пріоритетні тестові випадки у плани тестування. Гарна практика – мати полігон інформаційної мережі. Хай навіть «іграшковий», але він повинен бути. Тим більше, що знайти ресурси для кількох віртуальних, або хмарних серверів не так вже і важко, маючи під адмініструванням продакт інформаційної мережі.

- **Збереження, агрегація та аналіз даних**

Дані, які збираються під час моніторингу, повинні бути збережені в безпечних, надійних та доступних системах зберігання.

Дані повинні бути оброблені таким чином, щоб їх можна було коректно ідентифікувати та порівнювати між собою. Це включає нормалізацію (перетворення даних у стандартизований формат) та кореляцію (встановлення зв'язків між різними наборами даних) для отримання повної картини подій.

Дані, пов'язані з безпекою, повинні бути збережені в окремих сховищах даних, щоб уникнути можливості впливу на їх цілісність або зміну зловмисниками.

Сховища моніторингу, які отримують та обробляють ці дані, повинні існувати довше за джерела даних, щоб забезпечити постійність процесу моніторингу.

Системи зберігання даних повинні бути стійкими та надійними, щоб забезпечити неперервну роботу систем виявлення вторгнень.

Мережні журнали можуть бути докладними та займати дуже великі об’єми. Впровадьте різні рівні систем зберігання. З часом дані переносяться до так званого холодного зберігання (повільні дискові

сховища або інші, дешевші носії). Цей підхід є вигідним, оскільки старіші журнали потоку зазвичай не використовуються активно і потрібні лише на вимогу. Цей метод забезпечує ефективне керування сховищами, а також забезпечує доступ до історичних даних, коли вам це потрібно.

Потоки робочого навантаження зазвичай складаються з кількох джерел журналювання. Дані моніторингу необхідно ретельно аналізувати з усіх цих джерел. Наприклад, брандмауер блокуватиме лише трафік, який досягає його. Якщо у вас є група безпеки мережі, яка вже заблокувала певний трафік, цей трафік невидимий для брандмауера. Щоб реконструювати послідовність подій, потрібно агрегувати дані з усіх компонентів, які перебувають у потоці, а потім агрегувати дані з усіх потоків. Ці дані особливо корисні в сценарії реагування після інциденту, коли потрібно зрозуміти, що сталося. Дуже важливий точний відлік часу. З міркувань безпеки всі системи мають використовувати єдине мережне джерело часу, щоб вони завжди були синхронізовані.

- ✓ **Централізоване виявлення загроз із корельованими журналами.** Гарна практика, використання Security information and event management (SIEM), щоб консолідувати дані безпеки в центральному сховищі – базі даних, де їх можна корелювати між різними службами. Ці системи мають вбудовані механізми виявлення загроз. Вони можуть підключатися до зовнішніх каналів для отримання даних розвідки про загрози. Microsoft Defender Threat Intelligence, наприклад, публікує дані аналізу загроз, які можуть бути використані. Також можливо придбати канали аналізу загроз від інших постачальників, як-от Anomali ThreatStream та FireEye Helix. Ці канали можуть надати цінну інформацію та підвищити вашу безпеку.

  Крім SIEM ще деякі класи спеціалізованих систем з розвідки загроз та моніторингу, призначені саме для агрегації, аналізу та використання інформації з таких джерел.
  - Threat Intelligence Platforms
  - EDR (Endpoint Detection and Response).
  - Firewalls та IDS/IPS (Intrusion Detection/Prevention Systems)
  - SOAR (Security Orchestration, Automation, and Response)

  Система SIEM може генерувати сповіщення на основі корельованих і нормалізованих даних. Ці сповіщення є значним ресурсом під час процесу реагування на інциденти.

  Компроміс: системи SIEM можуть бути дорогими, складними та вимагати спеціальних навичок. Однак, якщо у вас відсутні

такі навички, потрібно буде обробляти та аналізувати дані SIEM самостійно. Це може бути трудомістким і складним процесом.

Системами SIEM зазвичай керує центральна команда організації. Якщо у вашій організації немає таких фахівців, подумайте про сторонню підтримку. Це може полегшити тягар ручного аналізу та кореляції журналів, щоб забезпечити більш ефективне та ефективне керування безпекою.

Комбінуючи кілька менших інструментів, ви можете емулювати деякі функції системи SIEM. Однак вам потрібно знати, що ці тимчасові рішення можуть не виконувати кореляційний аналіз. Ці альтернативи можуть бути корисними, але вони не можуть повністю замінити функціональність спеціальної системи SIEM, яка «заточена» під

- виявлення загроз,
- розслідування інцидентів,
- оцінку ризиків,
- управління кібербезпекою

- **Постійне виявлення та фіксація зловживань.**
  - ✓ **Будьте проактивними щодо виявлення загроз** і будьте пильні щодо ознак зловживань, як-от атаки методом (brute force) грубої обробки даних на компонент SSH або кінцеву точку RDP. Хоча зовнішні загрози можуть створювати багато шуму, особливо якщо програма працює в Інтернеті, внутрішні загрози повинні викликати більше занепокоєння. Несподівана атака грубої сили (brute force) з надійного мережного джерела або ненавмисна неправильна конфігурація, наприклад, повинні бути розслідувані негайно.
    Моніторинг не замінить проактивне зміцнення середовища. Більша площа поверхні схильна до більшої кількості атак. У міру практики посильте контроль. Наприклад
    - виявляйте та вимикайте невикористовувані облікові записи
    - видаляйте невикористовувані порти
    - використовуйте брандмауер веб-додатків.

    До цих базових правил проактивних дій щодо виявлення загроз також відносяться наступні “the best practical”:
    - встановлюйте складні паролі
      Вимагайте від користувачів встановлювати складні паролі з комбінацією великих і малих літер, цифр і спеціальних символів. Це зробить атаку методом грубої сили набагато складнішою.
    - забороніть використання стандартних портів

Забороніть користувачам використовувати стандартні порти для з'єднань, такі як порт 22 для SSH. Використання нестандартних портів може зменшити ймовірність успішної атаки.

- аудит безпеки
  Регулярно проводьте аудит безпеки системи для виявлення можливих слабких місць і потенційних загроз. Це допоможе забезпечити, що ваша система захищена від нових видів атак.
- встановіть двофакторну автентифікацію
  Встановіть двофакторну аутентифікацію для важливих облікових записів, щоб ускладнити заволодіння ними зловмисниками. Це дозволить захистити систему навіть у випадку, якщо пароль викрадений.
- регулярно оновлюйте ПЗ
  Важливо регулярно оновлювати всі програми та операційну систему до останніх версій, оскільки вони містять патчі безпеки, які закривають відомі уразливості.

✓ **Виявлення на основі сигнатур** може детально перевірити систему. Це передбачає пошук ознак або кореляції між діями, які можуть вказувати на потенційну атаку. Механізм виявлення може ідентифікувати певні характеристики, які вказують на конкретний тип атаки. Не завжди можливо безпосередньо виявити командний механізм атаки. Однак часто існують підказки або шаблони, пов'язані з певним процесом керування. Наприклад, на атаку може вказувати певна швидкість потоку з точки зору запиту, або вона може часто звертатися до доменів, які мають певні закінчення.

✓ Виявляйте **аномальні моделі доступу користувачів**, щоб ви могли ідентифікувати та досліджувати відхилення від очікуваних моделей. Це передбачає порівняння поточної поведінки користувача з минулою, щоб виявити аномалії. Хоча виконати це завдання вручну може бути неможливим, для цього можна скористатися інструментами аналізу загроз. Інвестуйте в інструменти Analytics and Entity Behavior Analytics (UEBA), які збирають поведінку користувачів із даних моніторингу та аналізують її. Ці інструменти часто можуть виконувати прогнозний аналіз, який зіставляє підозрілу поведінку з потенційними типами атак.

2. Використання системного та мережного моніторингу для виявлення вторгнень, зламів та інших загроз безпеці.

➢ **Типові загрози безпеці, які можна виявити за допомогою моніторингу:**

1. Вторгнення, або несанкціонований доступ до систем або мереж

   Спроби отримати доступ до конфіденційної інформації, системних ресурсів або послуг без дозволу.

   Методи моніторингу та виявлення:

   - **Системні журнали (логи):**
     - невдалі спроби входу
     - спроби доступу до неприпустимих ресурсів
     - виконання незвичайних команд
   - **Мережні монітори:**
     - невідомі підключення
     - аномальний обсяг трафіку
     - спроби перехоплення даних
   - **Системи IDS/IPS:** аналіз трафіку для виявлення ознак атак, автоматичне реагування.
   - **Моніторинг користувацької активності:** виявлення підозрілих дій, які можуть свідчити про компрометацію акаунтів.

2. Злам, або отримання несанкціонованого доступу до даних чи систем

   Одна з найбільш руйнівних загроз, коли зловмисник отримує доступ без авторизації.

   Методи моніторингу та виявлення:

   - **Системні журнали (логи):**
     - спроби незаконного доступу
     - невдалі спроби авторизації
   - **Моніторинг доступу до даних:** виявлення неповноважних спроб доступу.
   - **Системи IDS/IPS:** виявлення аномальної активності.
   - **Моніторинг користувачів:** аналіз незвичних патернів поведінки.
   - **Аналіз зловмисного ПЗ:** виявлення компрометації.

3. Шкідливе програмне забезпечення (віруси, трояни, шпигунське ПЗ, ботнети)

   Порушують конфіденційність, цілісність і доступність даних.

   Методи моніторингу та виявлення:

   - **Антивірусні програми:** виявлення вірусів і троянів.
   - **Мережний моніторинг:**
     - підозрілі з’єднання
     - контакти з центрами управління ботнетів
   - **Журнали подій:** аналіз підозрілої активності.
   - **Системи IDS:** виявлення мережних ознак зараження.
   - **HIDS (моніторинг виконання коду):** аналіз незвичайної активності.

4. Фішингові атаки

Використання підроблених листів, сайтів чи повідомлень для викрадення даних.

Методи моніторингу та виявлення:

- **Фільтрація електронних листів:** блокування спаму й небезпечних повідомлень.
- **Аналіз вмісту листів:** перевірка посилань, вкладень, запитів на персональні дані.
- **Моніторинг мережного трафіку:** виявлення підключень до підробних сайтів.
- **Системи антифішингу:** пошук індикаторів фішингових атак.
- **Освітні програми:** навчання користувачів розпізнавати фішинг.

5. DDoS-атаки

Перевантаження серверів чи мережі, що призводить до відмови в обслуговуванні.

Методи моніторингу та виявлення:

- **Мережний моніторинг:** аналіз великого обсягу трафіку та підозрілих запитів.
- **Системи IDS/IPS:** блокування небезпечних пакетів.
- **Моніторинг доступності сервісів:** контроль швидкодії та відмов.
- **Аналіз поведінки мережі:** виявлення аномальних патернів навантаження.
- **Системи швидкого реагування:** блокування атакуючого трафіку або його перенаправлення.

- **Конкретні приклади того, як моніторинг може допомогти виявити ці загрози:**

1. Моніторинг аномального завантаження процесора або використання пам’яті

Призначення: виявлення несподіваного навантаження, що може свідчити про діяльність шкідливого ПЗ або ботнетів.

Методи моніторингу та виявлення:

- Моніторинг ресурсів системи – відстеження використання процесора й пам’яті.
- Автоматичні попередження – сповіщення адміністраторів про різке зростання навантаження.
- Аналіз активності процесів – виявлення підозрілих або незвичайних дій.

2. Виявлення аномального мережного трафіку

Призначення: шкідливе ПЗ чи ботнети створюють надмірний обсяг трафіку.

Методи моніторингу та виявлення:

- Мережний моніторинг – відстеження обсягів трафіку.
- Системи IDS – аналіз мережної активності, виявлення незвичайних патернів.

3. Моніторинг підозрілої мережної активності

Призначення: виявлення атак та інших загроз через аналіз трафіку.

Методи моніторингу та виявлення:

- Моніторинг мережного трафіку – аналіз піків і спадів у навантаженні.
- Системи IDS – виявлення сканування портів, атак на протоколи.
- Моніторинг журналів подій мережних пристроїв – аналіз логів маршрутизаторів і комутаторів.
- Виявлення аномальних підключень – відстеження незвичайних з'єднань.
- Моніторинг протоколів мережі – аналіз використання протоколів і пошук підозрілих патернів.

4. Моніторинг несанкціонованого доступу до файлів або систем

Призначення: виявлення втручань у системи та дані.

Методи моніторингу та виявлення:

- Моніторинг змін у файлах – відстеження підозрілих модифікацій.
- Моніторинг журналів подій – аналіз невдалих входів та незвичних дій.
- Автоматичне сповіщення – повідомлення про зміну прав доступу, підозрілі спроби авторизації.
- Моніторинг активності користувачів – виявлення підозрілих дій з конфіденційними файлами.
- Системи IDS – аналіз спроб несанкціонованого доступу.

5. Моніторинг зловживання адміністративними правами

Призначення: контроль дій користувачів з розширеними повноваженнями.

Методи моніторингу та виявлення:

- Журнали подій системи – фіксація дій адміністраторів.
- Моніторинг змін у системі – контроль конфігурацій, прав доступу.
- Сповіщення про незвичайну активність – автоматичне попередження адміністраторів.
- Моніторинг дій з адмінправами – відстеження доступу до критичних ресурсів.

- Аудит доступу до системних ресурсів – перевірка спроб доступу з адмінправами.

6. Моніторинг поширення шкідливого програмного забезпечення

   Призначення: виявлення зараження систем і його поширення.

   Методи моніторингу та виявлення:

   - Моніторинг запуску файлів – виявлення підозрілих завантажень.
   - Системи IDS/IPS – виявлення аномальної мережної активності.
   - Моніторинг мережного трафіку – аналіз спроб з'єднання з небезпечними доменами.
   - Антивірусні системи – сканування файлів і блокування заражень.
   - Моніторинг системних ресурсів – контроль CPU, пам'яті, диску для виявлення аномалій.

7. Моніторинг фішингових атак

   Призначення: захист користувачів від підробних листів і посилань.

   Методи моніторингу та виявлення:

   - Моніторинг URL-адрес – аналіз підозрілих посилань.
   - Аналіз електронних листів – перевірка вмісту на ознаки шахрайства.
   - Використання спеціалізованих фільтрів – блокування небезпечних листів.
   - Сповіщення користувачів – автоматичні попередження про підозрілі листи.
   - Моніторинг поведінки користувачів – виявлення аномалій у використанні пошти.

8. Моніторинг DDoS-атак

   Призначення: забезпечення доступності сервісів під час атак перевантаження.

   Методи моніторингу та виявлення:

   - Моніторинг доступності сервісів – контроль роботи вебсайтів, поштових серверів, додатків.
   - Аналіз трафіку – виявлення аномальних піків у навантаженні.
   - Системи IDS/IPS – блокування атакованого трафіку.
   - Аналіз логів серверів – виявлення масових запитів із підозрілих джерел.
   - Моніторинг ресурсів – відстеження пропускної здатності мережі, CPU та пам'яті.

- **Рекомендації щодо налаштування моніторингу для виявлення загроз:**

- **Критичні системи** - це ті, без яких неможливо нормально функціонувати організації або іншій сутності. Це можуть бути системи управління базами даних, сервери електронної пошти, системи зберігання даних, мережні інфраструктури, системи моніторингу безпеки тощо.
  Критичні системи можуть також включати ті, які забезпечують забезпечення здоров'я та безпеку людей, наприклад, системи медичного обладнання, системи безпеки виробництва тощо.
- **Критичні дані** - це дані, які є найважливішими для діяльності організації або мають велику цінність для її функціонування. Це може бути конфіденційна інформація про клієнтів, фінансові дані, інтелектуальна власність, розробки, а також дані, які підлягають регулюванню законами про конфіденційність, такі як персональні дані.
- **Рекомендації щодо налаштування моніторингу**.
  - ✓ **Ідентифікація критичних активів**. Почніть з ідентифікації всіх критичних систем та даних в вашій організації. Це допоможе зосередити зусилля на захисті найважливішого.
  - ✓ **Налаштування моніторингу безпеки**. Встановіть системи моніторингу безпеки для постійного відстеження активності в мережі та на серверах. Це включає виявлення незвичайних патернів трафіку, спроби несанкціонованого доступу, аномальні активності користувачів тощо.
  - ✓ **Реагування на виявлені загрози**. Розробіть процедури та плани реагування на виявлені загрози. Це може включати автоматичні заходи безпеки, які активуються при виявленні підозрілих дій, а також ручні процедури для виявлення та усунення загроз вручну.
  - ✓ **Регулярне оновлення моніторингових систем**. Забезпечте, що ваші системи моніторингу постійно оновлюються та налаштовуються для виявлення нових видів загроз, оскільки кіберзлочинці постійно вдосконалюють свої методи.
  - ✓ **Навчання персоналу**. Проводьте навчання персоналу з питань кібербезпеки та впроваджуйте свідомість про безпеку в організації, щоб уникнути людських помилок, які можуть стати витоком для загроз безпеці.

Деякі рекомендації щодо вибору інструментів моніторингу з урахуванням потреб та бюджету:

- **Оцінка потреб**. Почніть з оцінки потреб вашої організації щодо моніторингу. Які типи систем і даних вам потрібно моніторити? Які загрози ви хочете виявляти? Які функції та можливості вам необхідні для ефективного моніторингу?

- **Аналіз бюджету.** Визначте ваш бюджет на моніторинг і безпеку. Розгляньте, скільки ви готові і можливі витратити на придбання та підтримку інструментів моніторингу.
- **Вибір відповідних інструментів.**
  - ✓ **Відкриті рішення.** Розгляньте використання відкритих рішень, таких як Nagios, Zabbix, OpenNMS для системного та мережного моніторингу. Вони зазвичай безкоштовні або мають відкритий код, що знижує витрати на ліцензії.
  - ✓ **Комерційні рішення**. Якщо ви готові витратити більше коштів, розгляньте комерційні рішення, такі як SolarWinds, Splunk, IBM QRadar. Вони можуть мати більше функціональності та підтримку, але можуть бути дорожчими.
  - ✓ **Спеціалізовані інструменти**. Для виявлення конкретних загроз, таких як DDoS-атаки або фішинг, ви можете розглянути використання спеціалізованих інструментів, таких як Cloudflare для захисту від DDoS-атак або PhishMe для виявлення фішингових спроб.
- **Управління ризиками.** Приймайте рішення про вибір інструментів, враховуючи рівень ризику вашої організації. Пам'ятайте про баланс між витратами та рівнем захисту, який вони забезпечують.
- **Тестування і оцінка.** Перед повним впровадженням інструментів проведіть тестування та оцінку їх ефективності. Впевніться, що вони відповідають вашим потребам і вимогам безпеки перед прийняттям остаточного рішення.

Необхідно згадати про таку річ, як налаштування правил та тригерів моніторингу для кожного типу загрози. Про більшість типів загроз ми вже говорили це

- **Вторгнення** (не санкціонований доступ до систем або мереж).
- **Зломи** (отримання не санкціонованого доступу до даних або систем).
- **Поширення шкідливого програмного забезпечення.**
- **Фішингові атаки.**
- **DDoS-атаки.**

Не згадували ще у цьому переліку такі неприємні речі, що дуже важко або навіть неможливо передбачити як

- **Вихід з ладу обладнання**
- **Стихійне лихо**.

Тут можливо лише рекомендувати використання систем моніторингу стану обладнання для виявлення незвичайних або непередбачених змін у функціонуванні та налаштування тригерів для сповіщень про великі зміни в енергоспоживанні або температурі обладнання, що може свідчити про проблеми з обладнанням або його виходом з ладу.

Не забуваємо, що процес моніторингу має бути безперервним, цілодобовим та супроводжуватися швидким реагуванням на інциденти.

Програмне забезпечення та бази даних загроз мають регулярно оновлюватися.

3. Аналіз логів, виявлення аномалій та паттернів, пов'язаних із зловмисними діями.

Класичний процес аналізу логів у контексті виявлення аномалій та патернів, пов’язаних із зловмисними діями, включає послідовність типових етапів обробки, інтерпретації та узагальнення даних. Узагальнене подання цих етапів наведено на рис. 8.01. Даний підхід дозволяє систематизувати процес аналізу логів і забезпечує більш ефективне виявлення потенційних загроз та відхилень у роботі системи.

1. **Фільтрація та збір та лог-даних (Collection of Log Data).** На цьому етапі відбувається пошук даних (в даному випадку – лог даних) з відповідних джерел. Велике значення має репрезентативність інформації, оскільки на цих даних буде оснований результат. Зважаючи на цей фактор велике значення має реальність та непошкодженість інформації. Зібранні дані мають бути поміщенні у деяке сховище (наприклад – база даних) та відфільтровані за відповідними критеріями використовуючи застосунки. Фільтрація даних має велике значення так як це впливає на репрезентативність даних.

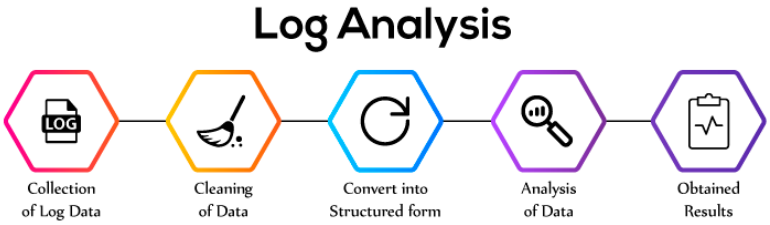


Рис. 8.01. Класичні етапи аналізу логів

2. **Структурування даних (Cleaning and Convert of Data into Structured form).** Дані логів можуть мати великий об'єм а також складну структуру. Для полегшення розуміння їх слід структурувати у зручному для користувача вигляді. Слід зважати на те, що логи можуть мати зв'язок з даними, що були отримані з інших джерел.
3. **Аналіз даних (Analysis of Data).** Завершаючим етапом є аналіз оброблених даних. На цьому етапі можуть бути залученні різні відомі методи аналізу даних. Вибір методу напряму залежить від типу задачі а також навиків розробника.

- **Аномалії та методики їх виявлення**

Виявлення аномалій відноситься до пошуку непередбачених значень (патернів) в потоках даних. Аномалія (викид, помилка, відхилення або виключення) - це відхилення поведінки системи від стандартного (очікуваного). Аномалії поділяються на три наступні категорії:

1. **Точкові аномалії** - виникають в ситуації, коли окремий екземпляр даних може розглядатися як аномальний по відношенню до решти даних. Даний вид аномалій є найбільш легко розпізнаваним, більшість існуючих методів створено для розпізнавання точкових аномалій.
2. **Контекстні аномалії**, також відомі як «умовні аномалії» – спостерігаються, якщо екземпляр даних є аномальним лише в певному контексті. Аномальна поведінка визначається за допомогою значень поведінкових атрибутів виходячи з конкретного контексту. Таким чином, екземпляр даних може бути контекстуальної аномалією за даних умов, але при таких же поведінкових атрибутах вважатися нормальним в іншому контексті.
3. **Колективні аномалії** – виникають, коли послідовність пов'язаних примірників даних (наприклад, ділянка часового ряду) є аномальною по відношенню до цілого набору даних. Окремий екземпляр даних в такій послідовності може не бути відхиленням, проте спільна поява таких екземплярів є колективною аномалією.

- **Режими виявлення аномалій**

Для вирішення завдання пошуку аномалій потребується набір даних, що описують систему. Кожен екземпляр в ньому описується міткою, яка вказує, чи є він нормальним або аномальним. Таким чином, безліч екземплярів з однаковими тегами формують відповідний клас.

Створення подібної промаркованної вибірки зазвичай проводиться вручну і є трудомістким і дорогим процесом. У деяких випадках отримати екземпляри аномального класу неможливо в силу відсутності даних про можливі відхилення в системі, в інших можуть бути відсутні мітки обох класів. Залежно від того, які класи даних використовуються для реалізації алгоритму, методи пошуку аномалій можуть виконуватися в одному з трьох перерахованих нижче режимів:

- ✓ **Supervised anomaly detection** (режим розпізнавання з учителем). Дана методика вимагає наявності навчальної вибірки, що повноцінно представляє систему і включає екземпляри даних нормального і аномального класів. Робота алгоритму відбувається в два етапи: навчання та розпізнавання. На першому етапі будується модель, з якою будуть порівнюються екземпляри, які не мають мітки. У більшості випадків покладається, що дані не змінюють свої статистичні характеристики, інакше виникає необхідність змінювати класифікатор. Основною складністю алгоритмів, що працюють в режимі розпізнавання з учителем, є формування даних для навчання.

Часто аномальний клас представлений значно меншою кількістю примірників, ніж нормальний, що може призводити до неточностей в отриманій моделі. У таких випадках застосовується штучна генерація аномалій.

- ✓ **Semi-Supervised anomaly detection** (режим розпізнавання частково з учителем). Вихідні дані при цьому підході представляють тільки нормальний клас. Навчившись на одному класі, система може визначати приналежність нових даних до нього, таким чином, визначаючи протилежний. Алгоритми, що працюють в режимі розпізнавання частково з учителем, не вимагають інформації про аномальний клас примірників, внаслідок чого вони ширше застосовуються й дозволяють розпізнавати відхилення за відсутності заздалегідь певної інформації про них.
- ✓ **Unsupervised anomaly detection** (режим розпізнавання без учителя). Застосовується при відсутності апріорної інформації про дані. Алгоритми розпізнавання в режимі без вчителя базуються на припущенні про те, що аномальні екземпляри зустрічаються набагато рідше нормальних. Дані обробляються, найбільш віддалені визначаються як аномалії. Для застосування цієї методики має бути доступний весь набір даних, тобто вона не може застосовуватися в режимі реального часу.

- ➢ **Системи моніторингу, що використовують аналіз логів.** У сучасних умовах інформаційної безпеки системи моніторингу, які використовують аналіз логів, відіграють ключову роль у виявленні загроз, інцидентів безпеки, а також у забезпеченні безперервного контролю за подіями в інфраструктурі. Журнали подій (логи) містять цінну інформацію про дії користувачів, системні процеси, мережну активність та події в додатках. Їхній своєчасний аналіз дозволяє виявити аномалії, що можуть свідчити про зловмисні дії або збої.

Такі системи збирають, зберігають, обробляють і візуалізують дані логів з різноманітних джерел – серверів, мережного обладнання, систем безпеки, додатків та ін. Деякі з найбільш відомих рішень у цій галузі:

- ❖ **ELK Stack (Elasticsearch, Logstash, Kibana)** – набір інструментів з відкритим вихідним кодом для централізованого збору, аналізу та візуалізації логів. До його складу входять:
  - **Elasticsearch** – високопродуктивна пошукова система, яка зберігає та індексує великі обсяги логів і дозволяє здійснювати швидкий пошук.
  - **Logstash** – система збору, фільтрації та обробки логів, яка підтримує велику кількість форматів і джерел даних.

- **Kibana** – потужний інструмент для створення дашбордів, графіків та візуального аналізу даних з Elasticsearch.

Це повністю безкоштовне рішення, однак воно потребує ручного встановлення, налаштування та технічної підтримки.

- ❖ **Graylog** – ще одна потужна система управління логами з відкритим кодом. Вона використовує Elasticsearch і MongoDB для зберігання даних та дозволяє здійснювати гнучкий пошук, фільтрацію, кореляцію та візуалізацію логів. Graylog зручний у налаштуванні, має інтуїтивний веб-інтерфейс і може масштабуватись для великих середовищ. Продукт має як безкоштовну версію (Community Edition), так і комерційні пакети з розширеною функціональністю.
- ❖ **Splunk** є одним з найпопулярніших комерційних рішень для обробки логів. Система пропонує високий рівень автоматизації, аналітики, створення звітів і кореляції подій. Splunk дозволяє не лише переглядати логи, а й будувати прогнози, створювати складні правила виявлення інцидентів безпеки, використовувати машинне навчання. Варто відзначити, що Splunk має безкоштовний рівень ліцензування, який дозволяє обробляти до 500 МБ логів на день без обмежень по функціональності.
- ❖ **SolarWinds LEM** – це комерційний продукт, який спеціалізується на моніторингу подій безпеки та логів у реальному часі. Він дозволяє автоматично реагувати на певні події, застосовувати правила кореляції, централізовано зберігати та аналізувати дані. SolarWinds LEM орієнтований на корпоративні мережі й інтегрується з іншими рішеннями цієї компанії для комплексного моніторингу інфраструктури.
- ❖ **IBM QRadar** – це потужна платформа типу SIEM (Security Information and Event Management), яка аналізує логи, мережний трафік, події безпеки та забезпечує комплексну аналітику інцидентів. Вона широко використовується в корпоративному та державному секторах. QRadar пропонує глибокий рівень кореляції подій, машинне навчання, підтримку MITRE ATT&CK та інтеграцію з рішеннями SOAR. Продукт є комерційним і досить вартісним, проте має високу ефективність у боротьбі з кіберзагрозами.
- ❖ **Wazuh** –сучасна безкоштовна SIEM-система з відкритим кодом, яка побудована на базі OSSEC. Вона підтримує збір логів, аналіз поведінки, контроль цілісності файлів, виявлення rootkit'ів, моніторинг політик безпеки. Має власну консоль керування, інтегрується з Elastic Stack, а також підтримує розширене управління користувачами та агенти для різних операційних систем.
- ❖ **Fluentd** – гнучкий, високопродуктивний лог-агрегатор з відкритим кодом**,** створений японською компанією Treasure Data (нині частина

Arm). Його основна мета — уніфікація збору та транспортування логів із різних джерел у централізоване сховище для подальшого аналізу.
Основні характеристики:

- Архітектура на базі плагінів — понад 500 плагінів, включно з джерелами (input), фільтрами, буферами й виходами (output). Це дозволяє легко інтегрувати Fluentd з такими системами як Elasticsearch, Kafka, Hadoop, Amazon S3, MongoDB, InfluxDB, Prometheus тощо.
- Підтримка JSON як формату за замовчуванням — Fluentd зберігає всі події як JSON-об'єкти, що забезпечує сумісність і простоту обробки.
- Підтримка буферизації, повторних спроб (retry) та failover — дуже важливо для роботи у розподілених системах, де тимчасова втрата доступу до сховища не призводить до втрати даних.
- Підтримка структурованих і неструктурованих логів — може бути використаний для логів додатків, вебсерверів, мережного обладнання, Docker/Kubernetes-подій тощо.

Використання у практиці:

- У **Kubernetes** середовищах часто використовується як агент (разом з Fluent Bit або самостійно) для збору логів із pod'ів та контейнерів.
- На відміну від Logstash, Fluentd вимагає менше ресурсів і є простішим у розгортанні.
- Активно застосовується в **DevOps, SRE та Cloud**-інфраструктурах як лог-транспортна система (часто в комбінації з ELK, Grafana, Prometheus).

Офіційний сайт https://www.fluentd.org/

❖ **SIEMonster** – модульна SIEM-платформа з відкритим кодом**,** яка побудована на основі найкращих безкоштовних інструментів кібербезпеки. Це комплексне рішення, розроблене спеціально для організацій, які хочуть мати SIEM-рівень захисту без високої вартості ліцензій Splunk, QRadar тощо.
Архітектура: SIEMonster — це не окремий продукт, а інтегрована платформа, яка складається з таких компонентів:

- **Elastic Stack** (Elasticsearch, Logstash, Kibana) — індексація та візуалізація логів.
- **Wazuh** — агент збору логів, виявлення атак, контроль цілісності файлів, реагування.
- **TheHive** — система управління інцидентами (SOC workflow).
- **Cortex** — механізм автоматизації реакцій на інциденти (playbooks, enrichment).
- **MISP** — платформа для обміну показниками компрометації (IoC).
- **Suricata** — IDS/IPS для моніторингу мережного трафіку.

Основні переваги:

- Повнофункціональна SIEM-система**,** яку можна розгорнути у приватній хмарі або на фізичних серверах.
- Готові Docker-композиції для швидкого розгортання.
- Можливість масштабування (агенти, обробка подій, інтеграція з будь-якими джерелами логів).
- Безкоштовна Community Edition для малих та середніх організацій.
- Гнучка рольова модель**,** орієнтована на побудову повноцінного SOC (Security Operations Center).

Цільова аудиторія:

- Організації, яким потрібне комплексне SIEM-рішення з інцидент-менеджментом, аналітикою, реагуванням.
- Університети, навчальні заклади, лабораторії — для вивчення архітектури повноцінної SIEM без великих витрат.
- Стартапи та MSSP (провайдери безпеки), які надають послуги моніторингу безпеки.

Недоліки:

- Потребує чимало ресурсів для повного розгортання.
- Наявність кількох взаємодіючих компонентів потребує доброго розуміння архітектури.
- Менше автоматизації з коробки порівняно з дорогими комерційними SIEM.

Офіційний сайт: https://siemonster.com/

Порівняння ключових особливостей:

Назва	**Тип ліцензії**	**Безкоштовна версія**	**Головні переваги**
ELK Stack	Відкритий код	Так	Гнучкість, масштабованість, візуалізація
Graylog	Відкритий код/ Комерційна	Так	Простота у використанні, інтеграція з Elasticsearch
Splunk	Комерційна	Так (до 500 МБ/день)	Машинне навчання, розширена аналітика
SolarWinds LEM	Комерційна	Ні	Автоматична кореляція подій
IBM QRadar	Комерційна	Ні	Високий рівень аналітики та безпеки
Wazuh	Відкритий код	Так	Моніторинг безпеки, інтеграція з ELK
Fluentd	Відкритий код	Так	Гнучкий збір логів у хмарних середовищах

Назва	Тип ліцензії	Безкоштовна версія	Головні переваги
SIEMonster	Відкритий код / Комерційна	Так	Інтегроване рішення з багатьма функціями безпеки

Системи аналізу логів є основою сучасного моніторингу, забезпечуючи виявлення проблем, вторгнень та аномалій ще до того, як вони вплинуть на функціонування інфраструктури. Вибір конкретного рішення залежить від розміру організації, технічної експертизи, потреб у функціональності та бюджету. Безкоштовні інструменти, як-от ELK Stack, Wazuh або Graylog, надають велику свободу та контроль, але потребують налаштування й обслуговування. Комерційні продукти, такі як Splunk чи QRadar, пропонують більше готових можливостей та аналітики, але потребують фінансових вкладень.

Важливо зазначити, що це лише декілька прикладів систем моніторингу, які використовують аналіз логів.

При виборі системи моніторингу важливо враховувати конкретні потреби та бюджет.

4. Розробка стратегій реагування та відновлення після виявлення загроз.

Більшість підприємств мають центральну групу безпеки, яка також відома як Security Operations Center (SOC) або SecOps. Основне завдання такої команди полягає у швидкому виявленні потенційних атак, визначенні їх пріоритетності та подальшому сортуванні інцидентів за рівнем загрози. Крім того, команда безпеки здійснює постійний моніторинг телеметричних даних, пов’язаних із безпекою, а також проводить детальне розслідування інцидентів і порушень.

Загальну схему процесу виявлення загроз та реагування на них у межах SOC наведено на рис. 8.02.

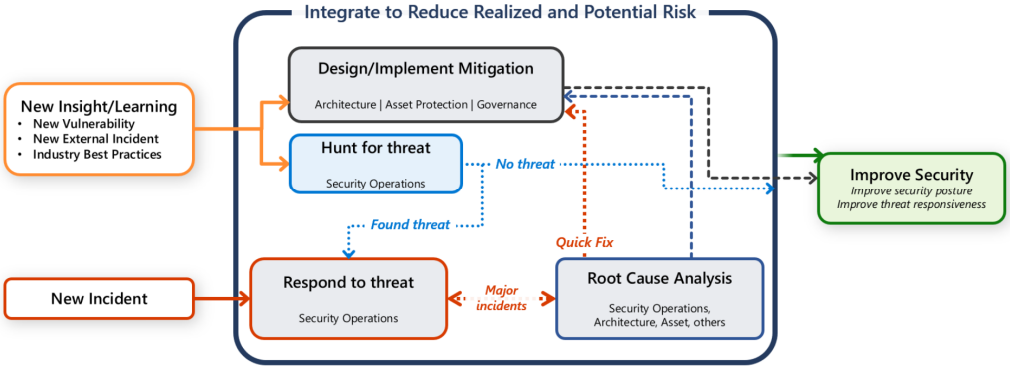


Рис. 8.02. Схема виявлення загроз та реакції на них.

Однак відповідальність за збереження робочого навантаження також покладається на відповідні сторони процесу. Важливо, щоб будь-які комунікації, розслідування та пошукові дії виконувалися як спільні зусилля команди робочого навантаження та команди SecOps.

Розглянемо рекомендації для робочої групи, які допоможуть швидко виявити, сортувати та розслідувати атаки.

Розглянемо термінологію, яка буде використоваватися:

Попередження	Повідомлення, що містить інформацію про інцидент.
Вірність попередження	Точність даних, які визначають сповіщення. Сповіщення високої точності містять контекст безпеки, необхідний для вжиття негайних дій. Сповіщення низької точності містять брак інформації або шум.
Хибно позитивний	Сповіщення, яке вказує на інцидент, якого не сталося.
Інцидент	Подія, яка вказує на неавторизований доступ до системи.
Реагування на інцидент	Процес, який виявляє ризики, пов'язані з інцидентом, реагує на них і пом'якшує їх.
Сортування	Операція з реагування на інциденти, яка аналізує проблеми безпеки та визначає пріоритетність їх пом'якшення.

- **Стратегія дизайну оповіщень.**

  Операції з реагування на інциденти виконуються, коли є сигнал або сповіщення про потенційну компрометацію. Сповіщення високої точності містять широкий контекст безпеки, який полегшує аналітикам прийняття рішень. Сповіщення високої точності призводять до низької кількості помилкових спрацьовувань. Необхідно передбачити, що система сповіщень фільтрує сигнали низької точності та зосереджується на сповіщеннях високої точності, які можуть вказувати на реальний інцидент.

  Ішними словами, стратегія дизайну оповіщень це як уникнути «сигнального шуму» та зосередитись на головному. Забезпечити, щоб тільки справді важливі, контекстуально обґрунтовані оповіщення доходили до аналітика SOC (Security Operations Center). Зовсім коротко - мінімізувати false positives і покращити швидкість реагування.

  Принципи якісного дизайну оповіщень

  - **Точність (Precision).** Оповіщення повинне з високою ймовірністю вказувати на справжню подію (наприклад, спробу компрометації). Важливо уникати тригерів, які можуть спрацьовувати при нормальній поведінці. Наприклад, замість «5 невдалих логінів» —

краще створити правило типу: «5 невдалих логінів + спроба входу з незвичної географії + підозрілий User-Agent».

- **Контекстність (Context-Rich Alerts).** Кожне сповіщення повинне містити дані, що дозволяють швидко зрозуміти ситуацію:
  - ✓ Ім'я користувача
  - ✓ IP-адреса джерела
  - ✓ Геолокація
  - ✓ Спроба входу в позаробочий час
  - ✓ Тип дій (read/write/delete)

  Чим більше контексту — тим менше часу на аналіз.
- **Пріоритезація та категоризація.** Кожне оповіщення має мати рівень критичності:
  - ✓ Low (інформаційне)
  - ✓ Medium (підозріле)
  - ✓ High/Critical (ймовірна атака)

  Система повинна мати фільтрацію та агрегацію за рівнем критичності.
- **Зменшення шуму (Noise Reduction).** Не можна, щоб аналітик отримував сотні малозначущих повідомлень щогодини. Методи:
  - ✓ **Aggregation** — об'єднувати схожі події в один інцидент.
  - ✓ **Suppression** — придушувати повторні події протягом певного часу.
  - ✓ **Thresholds** — сповіщення спрацьовує лише після перевищення порогу (наприклад, більше 20 невдалих логінів за 10 хвилин).

➢ **Налаштування сповіщень про інцидент.**

Сповіщення безпеки мають охопити відповідних людей у команді та організації. Встановлюється призначена контактна інформація у робочій групі, щоб отримувати повідомлення про інциденти. Ці сповіщення мають містити якомога більше інформації про ресурс, який зламано, і систему. Сповіщення має містити наступні дії, щоб команда могла пришвидшити дії.

Рекомендується реєструвати сповіщення про інциденти та дії та керувати ними за допомогою спеціальних інструментів, які зберігають контрольний слід. Використовуючи стандартні інструменти, необхідно зберегти докази, які можуть знадобитися для потенційних судових розслідувань. Шукаються можливості впровадження автоматизації, що може надсилати сповіщення на основі обов'язків відповідальних сторін. Зберігається чіткий ланцюжок зв'язку та звітування під час інциденту.

Широко використовуються переваги рішень для управління подіями безпеки (SIEM) і рішень для автоматизованого реагування оркестровки безпеки (SOAR), які надає організація. Крім того, можна придбати інструменти керування інцидентами та заохотити організацію стандартизувати їх для всіх робочих груп.

- **Розслідування інцидентів за допомогою групи реагування.**

Після виявлення загрози найважливіше провести ретельне розслідування, щоб зрозуміти її масштаби та наслідки. У цьому допоможе група реагування на інциденти (її ще називають командою оцінки).

Перший крок - створення групи реагування.

- ✓ Людина, яка отримала сповіщення про інцидент, повинна зібрати групу фахівців, виходячи з отриманої інформації.
- ✓ Цій групі потрібно узгодити методи та канали комунікації. Чи потрібні асинхронні обговорення (наприклад, листування) або термінові дзвінки? Як відстежуватимуть та повідомлятимуть про хід розслідування? Де знайти необхідні матеріали, пов'язані з інцидентом?

Ефективність розслідування залежить від актуальності документації.

- ✓ Важливо мати останні версії документів, що описують архітектуру системи, інформацію про компоненти, рівень конфіденційності та безпеки, власників компонентів та контактні особи. Застаріла інформація змусить групу витрачати дорогоцінний час на розуміння того, як працює система, хто відповідає за кожну її частину та які можуть бути наслідки інциденту.

До розслідування залучаються необхідні фахівці.

- ✓ Це може бути керівник реагування на інциденти, співробітник служби безпеки або відповідальні за окремі робочі навантаження.
- ✓ Не варто залучати людей, чиї обов'язки не стосуються конкретної проблеми. Іноді розслідування проводять окремі команди. Одна команда може первинно досліджувати проблему та намагатися її мінімізувати, а інша, більш спеціалізована, може проводити детальний цифровий аналіз для виявлення глибинних проблем. Для такої команди можна тимчасово ізолювати робоче середовище. В інших випадках усе розслідування може проводити одна команда.

На початковому етапі група реагування повинна визначити:

- ✓ **Можливий вектор атаки** та його вплив на **конфіденційність, цілісність та доступність (confidentiality, integrity, and availability - CIA)** системи.
- ✓ **Початковий рівень серйозності** інциденту, який враховує масштаби шкоди та терміновість виправлення. Цей рівень, ймовірно, змінюватиметься в міру отримання нових даних під час розслідування.

Під час розслідування потрібно визначити:

- ✓ **Необхідні дії та план інформування.** Чи потрібно змінити стан роботи системи? Як зупинити подальшу експлуатацію вразливості?

Чи потрібно інформувати внутрішні або зовнішні структури, наприклад, провести публічне розкриття інформації про вразливість?

- ✓ **Час виявлення та реагування.** Іноді законодавство зобов'язує вас повідомити про певні типи порушень регуляторні органи протягом певного часу (зазвичай годин або днів).

> Якщо прийнято рішення зупинити роботу системи, наступним кроком буде відновлення її працездатності за допомогою процедур аварійного відновлення. Якщо зупиняти роботу системи не потрібно, визначте, як усунути інцидент без впливу на її функціональність.

- **Відновлення після інциденту.**

До інцидентів кібербезпеки слід ставитися як до наслідків стихійного лиха, що потребують системного та контрольованого підходу до відновлення. У випадках, коли усунення наслідків інциденту вимагає повного перезапуску системи, застосовуються відповідні механізми відновлення після аварій (Disaster Recovery, DR), при цьому обов’язково враховуються вимоги інформаційної безпеки. Процес відновлення має бути організований таким чином, щоб запобігти повторному виникненню тієї ж проблеми: відновлення із зараженої резервної копії може призвести до повторного інфікування системи, а повторне розгортання середовища з невиправленою вразливістю — до повторного інциденту. У зв’язку з цим особлива увага приділяється перевірці всіх кроків і процедур переходу на резервну інфраструктуру, а також контрольованому поверненню до основної системи після завершення відновлення.

Якщо система продовжує працювати в нормальному режимі, насамперед оцінюється вплив інциденту на її загальне функціонування та ключові показники роботи. Паралельно продовжується моніторинг системи для контролю інших параметрів надійності та продуктивності з метою перевірки їх відповідності встановленим вимогам або, за необхідності, здійснюється їх коригування шляхом застосування відповідних процедур зниження негативного впливу. При цьому всі заходи реагування повинні реалізовуватися таким чином, щоб не порушувати конфіденційність оброблюваних даних.

Діагностика інциденту розглядається як інтерактивний і безперервний процес, який триває до моменту чіткого визначення вектора атаки, можливих варіантів усунення проблеми та сценаріїв відкату системи. Після завершення діагностичного етапу команда переходить до фази виправлення, яка передбачає визначення та

впровадження необхідного рішення у межах допустимого часу реагування.

Показники відновлення використовуються для вимірювання часу, необхідного для усунення інциденту та відновлення нормальної роботи системи. У випадках критичних збоїв або повної зупинки системи часові рамки виправлення можуть мати вирішальне значення. Для стабілізації роботи системи необхідно передбачити достатній час на застосування виправлень, встановлення патчів, проведення тестування та розгортання оновлень.

На завершальному етапі розробляється стратегія стримування, спрямована на запобігання подальшому поширенню інциденту та мінімізацію можливих збитків. Окремо формуються та впроваджуються процедури повного усунення загрози із системи, що забезпечує її безпечне відновлення та повернення до стабільного стану.

Компроміс: під час інциденту, ймовірно, не можливо дотримуватися інших функціональних або нефункціональних вимог. Наприклад, необхідно тимчасово відключити частини системи під час розслідування або навіть повністю зупинити її роботу, поки не буде визначено масштаби інциденту. Керівництво повинно чітко визначити прийнятні цілі під час інциденту та призначити відповідальну за це рішення особу.

- **Навчання на інцидентах.**

Інцидент кібербезпеки розкриває недоліки або вразливі місця в проєктуванні чи реалізації системи. Це можливість для покращення за рахунок уроків, отриманих з технічних аспектів проєктування, автоматизації, процесів розробки продуктів, що включають тестування, та ефективності реагування на інциденти.

Зберігайте детальні записи про інциденти, включаючи вжиті дії, хронологію та висновки.

Настійно рекомендується проводити структуровані огляди після інцидентів, такі як аналіз корінних причин та ретроспективи. Відстежуйте та пріоритезуйте результати цих оглядів і розгляньте можливість використання отриманих знань у майбутньому проєктуванні робочих навантажень.

Плани покращення повинні включати оновлення навчальних заходів з безпеки та тестування, наприклад, навчання з безперервності бізнесу та відновлення після аварій (BCDR). Використовуйте компрометацію безпеки як сценарій для проведення навчань з BCDR. Навчання можуть підтвердити, як працюють задокументовані процеси.

Не повинно бути кількох сценаріїв реагування на інциденти. Використовуйте єдине джерело, яке можна адаптувати залежно від

масштабу інциденту та його поширення. Навчання базуються на гіпотетичних ситуаціях. Проводьте навчання в середовищі з низьким рівнем ризику та включайте фазу навчання до самих навчань.

Проводьте огляди після інцидентів, або пост-mortem аналіз, щоб визначити слабкі місця в процесі реагування та сфери для покращення. На основі уроків, отриманих з інциденту, оновіть план реагування на інциденти (IRP) та засоби безпеки.

- **Інформування після інциденту**

Реагування на інцидент кібербезпеки обов'язково включає етап інформування всіх зацікавлених сторін. У першу чергу необхідно повідомити користувачів про виникнення збою в роботі системи, а також надати внутрішнім зацікавленим особам актуальну інформацію щодо виконуваних дій з відновлення та подальшого покращення системи. Окрім цього, слід інформувати співробітників організації про будь-які зміни в базових засобах безпеки робочого навантаження, що дозволяє зменшити ризик виникнення подібних інцидентів у майбутньому.

Для систематизації процесу інформування доцільно формувати окремий план комунікацій для різних цілей. Зокрема, передбачаються звіти про інциденти для внутрішнього використання, які документують перебіг подій та всі дії, виконані для їх усунення, а також звіти для регуляторного або юридичного дотримання, які готуються у випадках, коли цього вимагає законодавство або інші нормативні вимоги.

Для забезпечення ефективності процесу інформування рекомендується використовувати стандартизований формат звітності з чітко визначеними розділами, що дозволяє команді центру безпеки (SOC) оперативно формувати звіти для будь-яких інцидентів. Крім того, важливо забезпечити наявність пов'язаного звіту для кожного інциденту до моменту його закриття та завершення розслідування.

- **Рекомендації щодо інформування про небезпеки в інформаційних мережах**

Інформаційні мережі, як і будь-яка інша система, можуть бути вразливими до кіберзагроз. Системи моніторингу, такі як SIEM (Security Information and Event Management) та SOAR (Security Orchestration, Automation, and Response), відіграють важливу роль у виявленні та реагуванні на ці загрози.

SIEM (Security Information and Event Management) та SOAR (Security Orchestration, Automation, and Response) - це обидва важливі інструменти для кібербезпеки, але вони мають різні функції та підходи.

  - **SIEM (Security Information and Event Management)** – система, яка збирає, аналізує та інтерпретує дані з різних джерел у реальному часі з метою виявлення потенційних загроз та ризиків

для безпеки. SIEM зазвичай використовується для моніторингу подій, виявлення аномалій, аналізу журналів, зберігання даних та генерації звітів. Він може інтегруватись з іншими інструментами безпеки для покращення виявлення та реагування на загрози.

- **SOAR (Security Orchestration, Automation, and Response)** - це платформа, яка комбінує в собі засоби автоматизації, оркестрації та відповіді на події безпеки з метою ефективного управління інцидентами та зменшення часу реагування. SOAR допомагає організаціям у стандартизації та автоматизації процесів реагування на кіберзагрози. Це включає автоматизовану обробку інцидентів, інтеграцію з іншими інструментами безпеки, створення та виконання відповідних дій в разі інцидентів та відслідковування їх статусу.

Отже, основна різниця між SIEM і SOAR полягає в тому, що SIEM спрямований на моніторинг та аналіз подій безпеки, тоді як SOAR розвиває цей концепт, додаючи елементи автоматизації, оркестрації та відповіді на події.

- ✓ **Приклади SIEM** (Security Information and Event Management) рішень:
  - ❖ **Splunk Enterprise Security** використовується для збору даних з різних джерел, їх аналізу та кореляції для виявлення загроз.
  - ❖ **ArcSight ESM** використовується для централізованого моніторингу та управління журналами безпеки.
  - ❖ **IBM QRadar** використовується для виявлення аномалій та кореляції подій для виявлення загроз.
- ✓ **Приклади SOAR** (Security Orchestration, Automation, and Response) рішень:
  - ❖ **Demisto** використовується для автоматизації завдань реагування на інциденти.
  - ❖ **Palo Alto Networks Cortex XSOAR** використовується для оркестрації та автоматизації завдань кібербезпеки.
  - ❖ **IBM Resilient** використовується для автоматизації та координації реагування на інциденти.

Ці рішення можуть бути використані для:

- ❖ Збору даних з різних джерел, таких як журнали безпеки, мережні пристрої та агенти на кінцевих точках.
- ❖ Аналізу даних для виявлення загроз.
- ❖ Кореляції подій для визначення масштабу та впливу інциденту.
- ❖ Автоматизації завдань реагування на інциденти.
- ❖ Створення звітів про інциденти.

Вибір SIEM/SOAR рішення залежить від потреб вашої організації. Важливо враховувати такі фактори:

- ❖ Розмір вашої організації.
- ❖ Складність вашої мережі.
- ❖ Ваш бюджет.
- ❖ Ваші вимоги до безпеки.

Рекомендується проконсультуватися з фахівцем з кібербезпеки, щоб вибрати правильне рішення для вашої організації.

На основі даних моніторингу та рішень SIEM/SOAR можна сформувати наступні рекомендації щодо інформування про небезпеки в будь-якій інформаційній мережі:

1. **Створення та підтримка актуального плану інформування.**
   План повинен визначати, хто буде повідомлений про які інциденти, коли та яким чином. Необхідно чітко визначити ролі та відповідальності за інформування.
   План має бути регулярно переглядатись та оновлюватись, щоб враховувати зміни в мережі та загрозах.
2. **Використання каналів зв'язку, які відповідають потребам аудиторії.**
   Для технічних фахівців можуть бути більш корисними детальні технічні звіти.
   Керівництво може потребувати коротких оглядів з акцентом на бізнес-ризики та дії, які потрібно вжити. Для широкої аудиторії можуть бути корисними прості та зрозумілі повідомлення.
3. **Своєчасне інформування про всі інциденти.**
   Важливо негайно повідомити про критичні інциденти, щоб можна було вжити відповідних заходів для їхнього стримування. Про менш серйозні інциденти можна повідомляти з меншою терміновістю, але все ж таки вчасно, щоб дати можливість зацікавленим сторонам вжити заходів для захисту.
4. **Надання чіткої та лаконічної інформації.**
   Повідомлення про інциденти повинні містити яку інформацію:
   - Дата та час інциденту
   - Тип інциденту
   - Вплив на мережу та дані
   - Вжиті дії
   - Рекомендації для зацікавлених **сторін**
5. **Використання автоматизованих інструментів для інформування.**
   SIEM/SOAR рішення можуть автоматизувати процес надсилання повідомлень про інциденти, заощадити час та забезпечити своєчасне інформування всіх зацікавлених сторін.
6. **Проведення навчання з кібербезпеки для користувачів мережі.**

Навчання може допомогти користувачам розпізнавати та повідомляти про підозрілу активність а також може знизити ризик успішної атаки.

7. **Регулярний перегляд та оновлення плану інформування.**
   Важливо регулярно переглядати план інформування, щоб враховувати зміни в мережі, загрозах та потребах аудиторії. Це допоможе забезпечити ефективність інформування про інциденти.

Важливо пам'ятати, що ефективне інформування про інциденти є важливою складовою кібербезпеки.

5. Компроміси безпеки

Безпека повинна гарантувати конфіденційність, цілісність та доступність даних користувачів. Ніколи не можна йти на компроміси щодо заходів безпеки, але є моменти, що виглядають як компроміси безпеки, але насправді підвищують надійність мережі та її безпеку [10].

- **Компроміс безпеки з надійністю**
  Підвищена складність. Надійність це простота. Рекомендується звести до мінімуму точки відмови.
  - Деякі елементи керування безпекою можуть збільшити ризик неправильної конфігурації, що може призвести до збою служби. Приклади елементів керування безпекою, які можуть призвести до неправильної конфігурації, включають правила мережного трафіку, постачальників ідентифікаційних даних, виключення сканування вірусів і призначення контролю доступу на основі ролей або атрибутів.
  - Збільшення сегментації зазвичай призводить до більш складного середовища з точки зору топології ресурсів і мережі та доступу оператора. Ця складність може призвести до збільшення кількості точок збою в процесах і виконанні робочого навантаження.
  - Інструменти безпеки часто включені в багато рівнів архітектури інфраструктури. Ці інструменти можуть впливати на стійкість, доступність і планування потужностей. Неврахування обмежень у інструментах може призвести до зниження надійності, як-от виснаження порту NAT на вихідному брандмауері.

- **Збільшення критичних залежностей.**
  Надійність: Важливо мінімізувати критичні залежності компонентів інфраструктури, щоб мати більше контролю над точками відмови.
  Безпека: Для верифікації особи та дій потрібні ключові компоненти інфраструктури. Якщо ці компоненти недоступні або не працюють, система не може гарантувати безпеку.

Ризики: Недоступність або несправність компонентів інфраструктури може призвести до проблем з верифікацією та, як наслідок, до ризиків для безпеки.

Приклади критичних залежностей:

- Мережні брандмауери
- Списки відкликаних сертифікатів
- Сервер NTP
- Microsoft AD

- **Компроміси безпеки - складність відновлення після аварій.**

Компоненти інфраструктури мають надійно відновлюватися після будь-яких катастроф.

- Контроль безпеки може вплинути на цільовий час відновлення. Цей ефект може бути спричинений додатковими кроками, які необхідні для розшифровки резервних копій даних, або затримками оперативного доступу, створеними сортуванням надійності сайту.
- Самі елементи керування безпекою, наприклад секретні сховища та їхній вміст або захист від DDoS, мають бути частиною плану аварійного відновлення компонентів інфраструктури та мають бути перевірені за допомогою вправ відновлення.
- Вимоги щодо безпеки чи відповідності можуть обмежити варіанти розташування даних або доступ до резервних копій. Це може додатково ускладнити відновлення, оскільки навіть офлайн-копії можуть бути сегментовані.

- **Компроміси безпеки - підвищена швидкість змін.**

Часті зміни в системі можуть вплинути на її надійність.

Суворіша політика оновлень та патчів призводить до більшої кількості змін у виробничому середовищі системи. Ці зміни можуть виникати з таких джерел:

- Частіші оновлення коду додатків
- Збільшення routine patchів операційних систем
- Оновлення версій програмного забезпечення або платформ даних
- Застосування патчів постачальників до програмного забезпечення в середовищі
- Ротація ключів, облікових даних служб та сертифікатів

Таким чином часті зміни можуть призвести до тимчасових проблем із надійністю, поки система адаптується до нових налаштувань.

Порада:

- Ретельно тестуйте оновлення та патчі перед їх застосуванням у product-середовищі.
- Розгляньте можливість використання поетапного розгортання оновлень, щоб мінімізувати ризики.

- **Компроміси безпеки при оптимізації витрат**
  - **Збільшення інфраструктури** пов'язане з тим, що оптимізація витрат у системі часто передбачає пошук способів зменшення кількості та різноманітності інфраструктурних компонентів, а також підвищення щільності їх розміщення для більш ефективного використання ресурсів.

У контексті співвідношення безпеки та витрат слід враховувати, що деякі інфраструктурні компоненти або архітектурні рішення використовуються виключно для забезпечення захисту безпеки, включно з конфіденційністю, цілісністю та доступністю систем і даних. Хоча такі елементи підвищують загальний рівень безпеки середовища, вони водночас збільшують загальні витрати, а отже також потребують оптимізації з точки зору економічної ефективності.

До прикладів додаткових витрат, пов'язаних із забезпеченням безпеки, належать обчислювальні ресурси та мережна сегментація, що застосовується для ізоляції даних. Це може передбачати запуск окремих екземплярів, запобігання спільному розміщенню компонентів і, як наслідок, зниження щільності використання ресурсів.

Додатково витрати зростають через використання спеціалізованих засобів спостереження, таких як системи SIEM, які забезпечують агрегацію та аналіз загроз, а також через впровадження спеціалізованих мережних пристроїв або функцій, включно з брандмауерами та засобами захисту від розподілених атак типу відмови в обслуговуванні (DDoS).

До таких витрат також належать засоби класифікації даних, необхідні для визначення рівнів конфіденційності та типів інформації, а також спеціалізовані сховища або обчислювальні ресурси, що забезпечують шифрування даних у стані спокою та під час передачі, зокрема апаратні модулі безпеки (HSM) або механізми конфіденційних обчислень.

Окремо враховуються витрати на створення ізольованих тестових середовищ і використання інструментів тестування для перевірки ефективності засобів безпеки та виявлення раніше невиявлених прогалин у їхньому покритті.

Такі компоненти часто існують не лише у виробничому середовищі, а й у середовищах попереднього виробництва та відновлення після аварій, що додатково збільшує загальну складність та вартість інфраструктури.

- **Підвищений попит на інфраструктуру**

Оптимізація витрат:

- Оптимізація витрат має на меті зменшення попиту на ресурси для використання більш дешевих SKU (Stock Keeping Unit - унікальний код, який використовується для ідентифікації товару в системі інвентаризації), меншої кількості екземплярів або зниження споживання.

Безпека vs. Витрати:

- Преміум-SKU: Деякі заходи безпеки в хмарних та сервісах постачальників, які можуть покращити стан безпеки системи, можуть бути доступні лише в дорожчих SKU або рівнях.
- Зберігання журналів: Моніторинг безпеки високої точності та дані аудиту, що забезпечують широке охоплення, збільшують витрати на зберігання. Дані спостереження за безпекою також часто зберігаються протягом більш тривалого періоду, ніж це потрібно для звичайного оперативного аналізу.
- Збільшене споживання ресурсів: Вбудовані та локальні засоби безпеки можуть призвести до додаткової потреби в ресурсах. Шифрування даних в стані спокою та під час передачі також може збільшити попит. Обидва сценарії можуть потребувати більшої кількості екземплярів або більших SKU.

❖ **Підвищені витрати на процеси та експлуатацію**

Загальна вартість володіння:

- Витрати на робочі процеси персоналу є частиною загальної вартості володіння системою та враховуються при розрахунку рентабельності інвестицій. Оптимізація цих витрат є рекомендацією стовпа "Оптимізація витрат".

Безпека vs. Витрати:

- Більш комплексний та суворий режим управління патчами призводить до збільшення часу та коштів, витрачених на ці рутинні завдання. Це збільшення часто поєднується з очікуванням інвестування в готовність до позапланового застосування патчів для усунення вразливостей нульового дня.
- Суворіший контроль доступу для зниження ризику несанкціонованого доступу може призвести до більш складного управління користувачами та оперативного доступу.

1. Навчання та обізнаність щодо інструментів та процесів безпеки займають час працівників, а також
2. Витрати на матеріали, інструкторів та, можливо, навчальні середовища.
3. Дотримання нормативних актів може потребувати додаткових інвестицій в аудит та генерацію звітів про відповідність.
4. Планування та проведення навчань з реагування на інциденти безпеки потребує часу.

5. Необхідно виділити час для проектування та виконання рутинних та позапланових процесів, пов'язаних з безпекою, таких як ротація ключів або сертифікатів.
6. Перевірка безпеки SDLC зазвичай потребує спеціальних інструментів. Вашій організації може знадобитися платити за ці інструменти.
7. Приоритизація та усунення проблем, виявлених під час тестування, також займають час.
8. Залучення сторонніх фахівців з безпеки для проведення "білого" тестування або тестування, яке проводиться без знання внутрішньої роботи системи (іноді відомого як "чорне" тестування), включаючи тестування на проникнення, призводить до додаткових витрат.

- **Компроміси безпеки з операційною досконалістю**
  - **Ускладнення прозорості та зручності обслуговування** пов’язане з тим, що операційна досконалість вимагає такої архітектури системи, яка була б легкою для підтримки та забезпечувала достатній рівень спостереження. Найзручнішими для експлуатації вважаються ті архітектури, в яких усі залучені сторони мають максимально прозоре уявлення про її роботу.

У контексті співвідношення безпеки та прозорості слід враховувати, що безпека значною мірою базується на детальному веденні журналів подій. Такі журнали дають змогу виконувати точний аналіз роботи системи, виявляти відхилення від базових показників і реагувати на інциденти. Водночас вони можуть генерувати значні обсяги даних, що ускладнює отримання інформації, необхідної для оцінки надійності або продуктивності.

Питання відповідності та конфіденційності пов’язане з тим, що дотримання вимог щодо захисту даних передбачає маскування або редагування окремих фрагментів журналів, а інколи й значних обсягів табличної інформації. Це робиться для забезпечення конфіденційності, однак створює необхідність оцінювання того, як такі обмеження впливають на можливості спостереження, оповіщення та реагування на інциденти.

Складність спостереження також зростає через сегментацію ресурсів. Надійний поділ системи на сегменти вимагає додаткових механізмів розподіленого трасування та кореляції подій між сервісами для коректного відстеження трафіку. Окрім цього, збільшується обсяг обчислювальних ресурсів і даних, які необхідно підтримувати та аналізувати.

У сфері контролю доступу та реагування на інциденти враховується, що окремі засоби безпеки можуть навмисно

обмежувати доступ до системних компонентів. Під час інцидентів це може уповільнювати дії операторів, яким потрібен терміновий доступ. У зв'язку з цим у плануванні реагування на інциденти більший акцент робиться на попередній підготовці та регулярних навчаннях для забезпечення належної ефективності дій.

❖ **Зниження швидкості та підвищення складності** пов'язане з тим, що команди, які відповідають за систему, вимушені постійно оцінювати її продуктивність з метою поступового покращення якості, частоти та ефективності процесів доставки. При цьому ускладнення самої системи безпосередньо впливає на обсяг зусиль і рівень ризиків, що супроводжують її експлуатацію.

У контексті змін та безпеки застосування більш суворого контролю змін і процедур їх затвердження, спрямованих на зменшення ризику появи вразливостей, може сповільнювати розробку та впровадження нових функцій. Одночасно вимоги до регулярних оновлень безпеки та застосування патчів можуть збільшувати частоту розгортань. Додатково, наявність політики ручного затвердження в операційних процесах ускладнює автоматизацію цих процесів.

У сфері тестування безпеки результати перевірок можуть містити критичні висновки, які потребують пріоритизації, через що може відкладатися або блокуватися запланована робота.

Щодо ведення журналів та автоматизації, рутинні, позапланові та аварійні процеси можуть вимагати обов'язкового аудиторського логування для відповідності вимогам регуляторів. Таке ведення журналів робить виконання процесів більш жорстко регламентованим і менш гнучким.

У частині керування користувачами та автоматизації ускладнення виникає через зростання деталізації ролей і прав доступу, що підвищує складність адміністрування. Паралельно збільшення кількості рутинних операційних задач, пов'язаних із безпекою, таких як керування сертифікатами, призводить до розширення переліку процесів, які необхідно автоматизувати.

❖ **Координація зусиль** у системі пов'язана з тим, що операційна автономія дозволяє команді, яка мінімізує зовнішні контакти та перевірки, ефективніше контролювати власні операції та планування робіт. За таких умов зменшується залежність від зовнішніх узгоджень і підвищується керованість внутрішніх процесів.

Водночас зі зростанням зовнішніх вимог до відповідності, які можуть надходити як від вищої організації, так і від зовнішніх регуляторів чи партнерів, ускладнюється процес досягнення та підтвердження відповідності вимогам аудиторів. Це потребує

додаткових зусиль для документування та координації між різними сторонами.

Окремим аспектом є залучення спеціальних навичок у сфері безпеки, яких часто не мають команди, що безпосередньо відповідають за систему. Такі компетенції зазвичай надаються більшою організацією або зовнішніми підрядниками, що в обох випадках вимагає налагодження чіткої координації доступу, відповідальності та взаємодії між усіма учасниками процесу.

Також плани реагування на інциденти та вимоги відповідності часто передбачають наявність чітко визначених механізмів комунікації, зокрема для відповідального розкриття порушень. Такі плани мають бути інтегровані в загальну систему координації зусиль у сфері забезпечення безпеки, щоб забезпечити узгоджені та своєчасні дії всіх залучених сторін.

- **Компроміси безпеки з ефективністю продуктивності**
  - **Затримка та накладні витрати**

Ефективність продуктивності передбачає роботу системи з мінімальними затримками та оптимальним рівнем накладних витрат. Водночас впровадження засобів безпеки неминуче впливає на ці показники. Так, інструменти інспекційного контролю безпеки, зокрема брандмауери та фільтри вмісту, інтегруються безпосередньо у потоки даних, які вони захищають, що призводить до додаткової перевірки та, відповідно, збільшення затримки обробки запитів.

Окремо варто враховувати механізми керування доступом, оскільки кожен виклик керованого компонента потребує додаткової перевірки через системи ідентифікації та авторизації. Такі операції споживають обчислювальні ресурси і в окремих випадках можуть вимагати мережної взаємодії для підтвердження прав доступу.

Ще одним суттєвим фактором є шифрування, яке передбачає виконання операцій шифрування та розшифрування з використанням значних обчислювальних ресурсів. Час виконання таких операцій та навантаження на систему зростають залежно від складності алгоритмів, а також від використання високоентропійних та різноманітних ініціалізаційних векторів (IV).

Додатковий вплив на продуктивність має ведення журналів, оскільки збільшення обсягів логування може призводити до перевантаження системних ресурсів і зниження пропускної здатності мережі через необхідність передачі та обробки великих обсягів даних. Крім того, сегментація ресурсів часто супроводжується появою додаткових мережних переходів у межах архітектури системи, що також впливає на загальну продуктивність.

  - **Збільшення ризику неправильної конфігурації**

Стабільне досягнення цільових показників продуктивності значною мірою залежить від коректності та передбачуваності реалізації архітектурного проєктування. У той же час баланс між безпекою та продуктивністю може порушуватися через помилки або надмірність у конфігурації засобів захисту, що призводить до зниження ефективності роботи системи.

До типових прикладів таких конфігурацій належать надмірна складність, порядок і кількість правил брандмауера, що знижує їх ефективність, відсутність виключення критично важливих файлів із перевірки антивірусними засобами або системами контролю цілісності, що може спричиняти конфлікти блокування. Також негативний вплив має некоректне використання веб-брандмауерів із глибокою інспекцією пакетів у випадках, коли вона не є необхідною для конкретних мов або платформ, що обслуговуються системою.

Висновки до розділу

У сучасному світі кіберзагрози стають дедалі більш витонченими та небезпечними. Організаціям усіх розмірів важливо мати надійні системи безпеки для виявлення та реагування на ці загрози. Моніторинг є ключовим компонентом будь-якої стратегії кібербезпеки, оскільки він дозволяє організаціям відстежувати свою мережу та системи на наявність ознак компрометації.

- ❖ Системний та мережний моніторинг може використовуватися для виявлення вторгнень, зламів та інших загроз безпеці.
- ❖ Аналіз журналів може допомогти виявити аномалії та патерни, пов'язані із зловмисними діями.
- ❖ Розробка стратегій реагування та відновлення є важливою для мінімізації впливу успішної кібератаки.
- ❖ Компроміси безпеки можуть мати серйозні наслідки для організацій, включаючи фінансові втрати, шкоду репутації та перерви в роботі.

Моніторинг є важливим інструментом для виявлення та реагування на кіберзагрози. Організації повинні впровадити комплексні стратегії моніторингу, які включають системний та мережний моніторинг, аналіз журналів та розробку планів реагування та відновлення.

- ❖ Інвестуйте в рішення для моніторингу безпеки, які відповідають вашим потребам.
- ❖ Налаштуйте свої системи моніторингу для виявлення підозрілої активності.
- ❖ Регулярно переглядайте свої журнали та шукайте аномалії.
- ❖ Розробіть план реагування на інциденти та регулярно його тестуйте.

❖ Підтримуйте свої системи та програмне забезпечення в актуальному стані.

Моніторинг - це не одноразова подія. Це постійний процес, який потребує постійної уваги та вдосконалення. Організації, які серйозно ставляться до кібербезпеки, повинні зробити моніторинг пріоритетом.

Контрольні запитання

1) Які прояви загроз може виявити моніторинг?
2) Які бувають категорії аномалій даних?
3) Які типові загрози безпеці можна виявити за допомогою моніторингу?
4) Якими методами можна виявляти шкідливе програмне забезпечення?
5) Що означає моніторинг на різних рівнях?
6) Які існують рівні масштабу впливу подій у моніторингу?
7) Якими методами можна виявляти фішингові атаки?
8) Для чого можуть використовуватися рішення SIEM та SOAR?
9) Які системи моніторингу використовують зовнішні канали для отримання даних розвідки про загрози?
10) Яку інформацію повинно містити повідомлення про інцидент?
11) Які порушення ідентифікації та доступу підлягають моніторингу?

Джерела та посилання

1. Amazon Web Services. Amazon CloudWatch User Guide. AWS Documentation, 2025. – [Online]. – Режим доступу: https://docs.aws.amazon.com/cloudwatch
2. Anjaria, A. Introduction to System Monitoring [Електронний ресурс] // Medium. – 2025. – Режим доступу: https://medium.com/@anasanjaria/introduction-to-system-monitoring-02946f967ba3
3. Akiya, N., et al. MPLS Segment Routing Label Type for IPFIX (RFC 9160) [Електронний ресурс]. – 2022. – Режим доступу: https://www.rfc-editor.org/info/rfc9160
4. BetterStack. SRE Fundamentals: Metrics, Monitoring, and Alerting [Електронний ресурс]. – 2025. – Режим доступу: https://betterstack.com/community/guides/monitoring/what-is-metrics-monitoring-alerting/
5. Bhamare, D., Salman, O., Erbad, A., Jain, R. Network data analytics and visualization: A survey. IEEE Access, vol. 12, pp. 76543–76562, 2024.
6. Boucadair, M., Claise, B., Palet, R. New IPFIX Information Elements for TCP Options and IPv6 Extension Headers (RFC 9740) [Електронний ресурс]. – 2025. – Режим доступу: https://www.rfc-editor.org/info/rfc9740
7. Boucadair, M., Claise, B., Pentikousis, K., et al. Simple Fixes to the IP Flow Information Export (IPFIX) IANA Registry (RFC 9710) [Електронний ресурс]. – 2025. – Режим доступу: https://www.rfc-editor.org/info/rfc9710
8. Canonical. Monitoring Ubuntu Systems with Landscape [Електронний ресурс]. – 2023. – Режим доступу: https://ubuntu.com/landscape/docs/monitoring
9. Cisco Systems. NetFlow Version 5 Export Protocol — Overview & Configuration (IOS/IOS XE) [Електронний ресурс]. – 2024. – Режим доступу: https://www.cisco.com/…/netflow-v5-export-protocol.html
10. Datadog. Datadog documentation: dashboards & visualization. Datadog Docs, 2025. – Режим доступу: https://docs.datadoghq.com
11. Datadog Documentation. Infrastructure Monitoring [Електронний ресурс]. – 2024. – Режим доступу: https://docs.datadoghq.com/infrastructure/
12. East African Journal of Information Technology. Real-Time Monitoring of Network Devices: Its Effectiveness in Enhancing Network Security [Електронний ресурс]. – 2021. – Режим доступу: via ResearchGate
13. Elastic N.V. Elastic Observability Documentation (ELK Stack for Logs, Metrics, and APM) [Електронний ресурс]. – 2023. – Режим доступу: https://www.elastic.co/guide/en/observability/
14. Elastic. Security | Elastic Docs. – Режим доступу: https://www.elastic.co/docs/deploy-manage/security (дата звернення: 25.08.2025)
15. European Union Agency for Cybersecurity (ENISA). Guidelines on network monitoring and data visualization. ENISA Report, 2023. – Режим доступу: https://www.enisa.europa.eu
16. Filsfils, C., Leddy, J., Voyer, D., et al. Information Elements for IPv6 Segment Routing Network Programming (SRv6) (RFC 9487) [Електронний ресурс]. – 2023. – Режим доступу: https://www.rfc-editor.org/info/rfc9487
17. Fischer, J. Zabbix 6 IT Infrastructure Monitoring Cookbook. Packt Publishing, 2022. — Chapters 2–4.
18. GeeksforGeeks. Distributed Systems Monitoring [Електронний ресурс]. – 2025. – Режим доступу: https://www.geeksforgeeks.org/system-design/distributed-systems-monitoring/

19. Google Cloud. Cloud Monitoring dashboards. Google Cloud Documentation, 2024. – Режим доступу: https://cloud.google.com/monitoring/dashboards
20. Google SRE. Monitoring Distributed Systems — The Four Golden Signals [Електронний ресурс] // SRE Book. – Режим доступу: https://sre.google/sre-book/monitoring-distributed-systems/
21. Google SRE. Monitoring Systems with Advanced Analytics [Електронний ресурс] // SRE Workbook. – Режим доступу: https://sre.google/workbook/monitoring/
22. Hausenblas, M. Cloud Observability in Action. 1st ed., Manning Publications, 2023. — Chapters 1–2.
23. Icinga Blog. What a Cloud Monitoring Architecture looks like [Електронний ресурс]. – 2024. – Режим доступу: https://icinga.com/blog/what-a-cloud-monitoring-architecture-looks-like/
24. InMon. sFlow Agent Configuration and Troubleshooting Guide [Електронний ресурс]. – 2024. – Режим доступу: https://sflow.net/inmon/doc/HTML/Config[sFlow_Agent].htm
25. InMon / sFlow.org. Deploy real-time network dashboards using Docker compose (sFlow-RT) [Електронний ресурс]. – July 2024. – Режим доступу: https://blog.sflow.com/2024/07/
26. Infraon Blog. Importance of Network Performance Monitoring Tools 2025 [Електронний ресурс]. – 18 квітня 2022. – Режим доступу: https://infraon.io/blog/network-performance-monitoring-tools/
27. IR Collaborate. Network Monitoring 101: Tools, Metrics, and Best Practices [Електронний ресурс]. – 2025. – Режим доступу: https://www.ir.com/guides/network-monitoring
28. IJCA Online. The Strategic Role of Monitoring in Modern Critical Infrastructure Management [Електронний ресурс]. – 29 травня 2025. – Режим доступу: https://ijcaonline.org/archives/volume187/number8/the-strategic-role-of-monitoring-in-modern-critical-infrastructure-management/
29. LogicMonitor. Understanding the 5 Key Concepts of Network Monitoring [Електронний ресурс]. – 2025. – Режим доступу: https://www.logicmonitor.com/blog/network-monitoring-key-concepts
30. Majors, C., Fong-Jones, L., Miranda, G. Observability Engineering. O'Reilly Media, 2022. — Chapters 1–3.
31. Microsoft. Рекомендації щодо моніторингу та виявлення загроз для Power Platform робочих навантажень. – Режим доступу: https://learn.microsoft.com/uk-ua/power-platform/well-architected/security/monitor-threats (дата звернення: 25.08.2025)
32. Microsoft. System Center – Operations Manager 2025. Офіційна документація. – Режим доступу: https://learn.microsoft.com/en-us/system-center/scom/?view=sc-om-2025 (дата звернення: 25.08.2025)
33. Modern Network Observability: A hands-on approach. Packt, 2022. — Chapters 1–4.
34. Nagios Enterprises. Nagios Core Documentation: Monitoring Basics [Електронний ресурс]. – 2024. – Режим доступу: https://www.nagios.org/documentation/
35. Pandora FMS Blog. Distributed systems: features and best practices [Електронний ресурс]. – 2024. – Режим доступу: https://pandorafms.com/blog/types-of-distributed-systems/
36. Pradhan, R. Cloud Observability in Practice: Monitoring, Tracing, and Logging for Modern Applications. Packt Publishing, 2023. — Chapters 1–4.

37. Prometheus Authors. Prometheus Documentation: Architecture, Exporters, Alertmanager [Електронний ресурс]. – 2025. – Режим доступу: https://prometheus.io/docs/
38. Red Hat. Monitoring and Metrics in OpenShift (v4.15) [Електронний ресурс]. – 2024. – Режим доступу: https://docs.openshift.com/container-platform/4.15/monitoring/
39. Red Hat Documentation. Monitoring architecture [Електронний ресурс] // Red Hat OpenStack Platform Monitoring Tools Configuration Guide. – Режим доступу: https://docs.redhat.com/en/documentation/red_hat_openstack_platform/16.1/html/monitoring_tools_configuration_guide/architecture_osp
40. ResearchGate. Present and Future of Network Security Monitoring [Електронний ресурс]. – 2025. – Режим доступу: https://www.researchgate.net/publication/350164255_Present_and_Future_of_Network_Security_Monitoring
41. ResearchGate. Tools for Distributed Systems Monitoring [Електронний ресурс]. – 2025. – Режим доступу: https://www.researchgate.net/publication/311863266_Tools_for_Distributed_Systems_Monitoring
42. Sematext. Introduction to Metrics, Monitoring and Alerting [Електронний ресурс]. – 2025. – Режим доступу: https://sematext.com/blog/monitoring-alerting/
43. selector.ai. Network Monitoring in 2025: Techniques, Challenges, and How AI Can Help [Електронний ресурс]. – 9 липня 2025. – Режим доступу: https://www.selector.ai/blog/network-monitoring-in-2025-techniques-challenges-and-how-ai-can-help/
44. Snort Users Manual 2.9.16. – Режим доступу: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/ (дата звернення: 25.08.2025)
45. S. Newman. Building Microservices, 2nd ed., O'Reilly Media, 2021. — Part II: Monitoring.
46. SearchInform. System Monitoring: A Complete Guide for Modern IT [Електронний ресурс]. – 2025. – Режим доступу: https://searchinform.com/articles/cybersecurity/measures/security-monitoring/system-monitoring/
47. TechRadar. Best Network Monitoring Tools for 2025 [Електронний ресурс]. – серпень 2025. – Режим доступу: https://www.techradar.com/best/best-network-monitoring-tools
48. Turnbull, J. Monitoring with Prometheus and Grafana: Build High Performance Systems for Monitoring, Metrics, and Alerting, 2nd ed., O'Reilly Media, 2022. — Chapters 1–3.
49. Wikipedia. Network detection and response (NDR) [Електронний ресурс]. – 2025. – Режим доступу: https://en.wikipedia.org/wiki/Network_detection_and_response
50. Wikipedia. Network monitoring [Електронний ресурс]. – 2025. – Режим доступу: https://en.wikipedia.org/wiki/Network_monitoring
51. Wikipedia. Passive monitoring [Електронний ресурс]. – 2023. – Режим доступу: https://en.wikipedia.org/wiki/Passive_monitoring
52. Wikipedia. Prometheus (software) [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Prometheus_%28software%29
53. Wikipedia. Shinken (software) [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Shinken_%28software%29

54. Wikipedia. Simple Network Management Protocol (SNMP) [Електронний ресурс]. – 2025. – Режим доступу: https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol
55. Wazuh Documentation. Офіційна документація Wazuh. – Режим доступу: https://documentation.wazuh.com/current/index.html (дата звернення: 25.08.2025)
56. Splunk Documentation. Офіційна документація Splunk. – Режим доступу: https://docs.splunk.com/Documentation (дата звернення: 25.08.2025)
57. IBM QRadar SIEM 7.4.3 Documentation. Офіційна документація IBM QRadar SIEM 7.4.3. – Режим доступу: https://www.ibm.com/docs/en/qsip/7.4?topic=SS42VS_7.4/com.ibm.qradar.doc/c_qradar_pdfs.htm (дата звернення: 25.08.2025)
58. Grafana Labs Documentation. Grafana Cloud and OSS Monitoring Tools [Електронний ресурс]. – 2024. – Режим доступу: https://grafana.com/docs/
59. Grafana Labs. Grafana documentation. Grafana Labs, 2025. – Режим доступу: https://grafana.com/docs/
60. N. Diagboya. Infrastructure Monitoring with Amazon CloudWatch. Packt Publishing, 2021. — Chapters 1–2.
61. J. Riedesel. Software Telemetry. Manning Publications, 2025. — Chapters 2–4.
62. M. Hausenblas. Cloud Observability in Action. 1st ed., Manning Publications, 2023. — Chapters 1–2.

Навчальне видання

ОСНОВИ СИСТЕМНОГО ТА МЕРЕЖНОГО МОНІТОРИНГУ

Навчальний посібник

Підготував

Фальковський Ігор Геннадійович

Формат 60×84/16.
Ум. друк. арк. 7,32.
Безкоштовно

Редактура та комп’ютерне верстання авторські.
Гарнітура Orchidea Pro Md, Mariupol, Midpoint Pro Regular, Ubuntu Mono

---