**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ**
**КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ**

# КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Рекомендації щодо виявлення та реагування на інциденти безпеки в інформаційній системі організації за допомогою Wazuh»**

зі спеціальності *125 Кібербезпека*
*(код, найменування спеціальності)*

освітньо-професійної програми *Інформаційна та кібернетична безпека*
*(назва програми)*

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

_____ Олег ШАПКО
*(підпис)*

Виконав: здобувач(ка) вищої освіти групи БСД-42
ШАПКО Олег
*(прізвище, ім'я)*

Керівник к.військ.н., доц. ГАХОВ Сергій
*(науковий ступінь, вчене звання, прізвище, ім'я)*

Рецензент _____
*(науковий ступінь, вчене звання, прізвище, ім'я)*

Київ 2025

**ЗМІСТ**

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – операційна система

ПК – персональний комп’ютер

API – Application Programming Interface

APT – Advanced Persistent Threat

DDoS – Distributed Denial of Service

DNS – Domain Name System

EDR – Endpoint Detection and Response

IaaS – Infrastructure as a Service

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

SIEM – Security Information and Event Management

SOAR – Security Orchestration Automation and Response

SOC – Security Operations Center

UEBA – User and Entity Behavior Analytics

# ВСТУП

*Актуальність дослідження.* Сучасний ландшафт загроз характеризується високою динамікою, складністю, різноманітністю та постійною еволюцією. Організаціям необхідно бути пильними, постійно оновлювати свої знання про нові загрози та вживати комплексних заходів для захисту своїх інформаційних систем.

Виявлення та реагування на інциденти безпеки є невід'ємною частиною забезпечення інформаційної безпеки організації, що допомагає захистити її активи, підтримувати безперервність бізнесу, дотримуватися законодавчих вимог та зберігати довіру зацікавлених сторін. Ефективне реагування на інциденти дозволяє швидко локалізувати загрозу, відновити працездатність систем та мінімізувати час простою. Це допомагає організації продовжувати свою діяльність без значних перерв.

Інформаційні інциденти можуть завдати значної шкоди репутації організації та призвести до втрати довіри клієнтів, партнерів та інвесторів. Швидке та ефективне реагування на інциденти демонструє, що організація серйозно ставиться до безпеки та захисту даних, що може допомогти зберегти довіру. Процес реагування на інциденти часто включає аналіз причин інциденту, що може допомогти виявити слабкі місця та вразливості в інформаційній системі організації. Це дозволяє вжити заходів для їх усунення та запобігання майбутнім інцидентам.

Вищесказане визначає актуальність теми даної кваліфікаційної роботи, основний зміст якої становлять дослідження методів та засобів виявлення та реагування на інциденти безпеки в інформаційній системі організації на базі Wazuh.

*Об'єкт дослідження* – виявлення та реагування на інциденти безпеки в інформаційній системі організації.

*Предмет дослідження* – методи та засоби виявлення та реагування на

інциденти безпеки в інформаційній системі організації на базі Wazuh.

*Мета роботи* – розробити рекомендації щодо виявлення та реагування на інциденти безпеки в інформаційній системі організації на базі Wazuh.

*Наукові завдання:*

дослідити сутність проблеми виявлення та реагування на інциденти безпеки в інформаційній системі організації;

проаналізувати підходи до виявлення та реагування на інциденти безпеки в інформаційній системі організації;

проаналізувати існуючі рішення для виявлення та реагування на інциденти безпеки в інформаційній системі організації;

проаналізувати методи та засоби виявлення та реагування на інциденти безпеки в інформаційній системі організації на базі Wazuh;

розкрити порядок виявлення та реагування на інциденти безпеки в інформаційній системі організації на базі Wazuh.

*Методи дослідження* – опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння.

*Практичне значення одержаних результатів:* запропоновано порядок виявлення та реагування на інциденти безпеки в інформаційній системі організації на базі Wazuh, а також розроблено рекомендації фахівцям з кібербезпеки щодо їх реалізації.

# 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ІНЦИДЕНТИ БЕЗПЕКИ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ

## 1.1. Дослідження проблеми виявлення та реагування на інциденти безпеки в інформаційній системі організації

У Звіті ENISA Threat Landscape 2024 [1] зазначається, що протягом другої половини 2023 року та першої половини 2024 року спостерігалася помітна ескалація атак на кібербезпеку, що встановило нові стандарти як у різноманітності, так і в кількості інцидентів, а також у їхніх наслідках. Регіональні конфлікти, що тривають, все ще залишаються значним фактором, що формує ландшафт кібербезпеки. Феномен хактивізму неухильно поширювався, відзначаючись появою численних нових груп. Великі події, що відбувалися на національному чи європейському рівнях, стали мотивацією для посилення хактивістської активності (наприклад, європейські вибори) [1].

У Звіті ENISA Threat Landscape 2024 [1] висвітлюється та виділяється вісім основних типів загроз. Ці конкретні типи загроз були виділені через їх популярність протягом багатьох років, їх широке поширення та значний вплив у результаті реалізації цих загроз [1]:

*програми-вимагачі*. За даними ENISA [1] програма-вимагач визначається як тип атаки, коли суб’єкти загрози беруть під контроль активи цілі та вимагають викуп в обмін на повернення доступності активу або в обмін на оприлюднення даних цілі. Це визначення потрібне для охоплення мінливого ландшафту загроз програм-вимагачів, поширеності різноманітних методів вимагання та різноманітних цілей зловмисників, крім виключно фінансової вигоди. Програмне забезпечення-вимагач стало однією з головних загроз з кількома резонансними і набувшими широко розголосу інцидентами;

*шкідливе програмне забезпечення*. Шкідливе програмне забезпечення, яке також називають шкідливим кодом і шкідливою логікою, – це загальний термін,

який використовується для опису будь-якого програмного забезпечення або вбудованого програмного забезпечення, призначеного для виконання несанкціонованого процесу, який матиме негативний вплив на конфіденційність, цілісність або доступність системи;

*соціальна інженерія*. Соціальна інженерія охоплює широкий спектр діяльності, спрямованої на використання людських помилок або людської поведінки з метою отримання доступу до інформації чи послуг. Вона використовує різні форми маніпуляції, щоб обманом змусити жертв зробити помилку або передати конфіденційну або секретну інформацію. Користувачів можуть спонукати відкрити документи, файли чи електронні листи, відвідати веб-сайти або надати доступ до систем чи послуг. Незважаючи на те, що використовувані приманки та трюки можуть зловживати технологіями, вони покладаються на людський фактор для досягнення успіху. Соціальна інженерія складається в основному з таких векторів атак: фішинг, spear-phishing, whaling, smishing, vishing, watering hole attack, nataking, pretexting, quid pro quo, honeytraps і scareware. Хоча методи соціальної інженерії часто використовуються для отримання початкового доступу, вони також можуть використовуватися на наступних етапах інциденту чи порушення. Яскравими прикладами є компрометація бізнес-електронної пошти (BEC), шахрайство, видавання себе за іншу особу, підробка та, останнім часом, вимагання;

*загрози щодо даних*. Порушення даних визначається в GDPR як будь-яке порушення безпеки, що призвело до випадкового або незаконного знищення, втрати, зміни або несанкціонованого розкриття або доступу до персональних даних, що передаються, зберігаються або обробляються іншим чином (стаття 4.12 GDPR). Технічно кажучи, загрози для даних можна в цілому класифікувати як порушення даних або витік даних. Хоча вони часто використовуються як взаємозамінні, вони передбачають принципово різні концепції, які здебільшого залежать від того, як вони відбуваються.

Порушення даних – це навмисна кібератака, здійснена кіберзлочинцем з метою отримання несанкціонованого доступу та видачі конфіденційних,

конфіденційних або захищених даних.

Іншими словами, порушення даних – це навмисна атака на систему чи організацію з наміром викрасти дані. Витік даних – це подія (наприклад, неправильна конфігурація, вразливість або помилка людини), яка може спричинити ненавмисну втрату або розкриття чутливих, конфіденційних або захищених даних (навмисні атаки іноді називають розкриттям даних);

*загрози доступності: відмова в обслуговуванні*. DDoS атаки спрямовані на доступність системи та даних, і хоча це не нова загроза, вони відіграє значну роль у ландшафті загроз кібербезпеці. Атаки відбуваються, коли користувачі системи або служби не можуть отримати доступ до відповідних даних, послуг або інших ресурсів. Це може бути досягнуто шляхом виснаження служби та її ресурсів або перевантаженням компонентів мережевої інфраструктури. Вплив DDoS-атак часто обмежений і символічний;

*маніпулювання інформацією*. Маніпулювання інформацією та втручання (FIMI) описує здебільшого не протизаконну модель поведінки, яка загрожує або може негативно вплинути на цінності, процедури та політичні процеси. Така діяльність носить маніпулятивний характер, здійснюється цілеспрямовано та скоординовано. FIMI може здійснюватися державними чи недержавними суб’єктами, включаючи їх довірених осіб на території та за її межами.

На рисунку 1.1 можна побачити, що програми-вимагачі та DDoS-атаки були формами атак, про які найчастіше повідомлялося, і на них припадало більше половини спостережуваних подій, за якими йшли загрози, пов’язані з даними [1]. У кількох випадках інциденти стосувалися більш ніж однієї категорії загроз і, таким чином, аналізувалися в контексті всіх відповідних категорій. З огляду на те, що Звіт ENISA [1] базується на загальнодоступній інформації та той факт, що така інформація не завжди може надати повну картину, у деяких випадках інциденти не можуть бути віднесені до жодної категорії загрози.

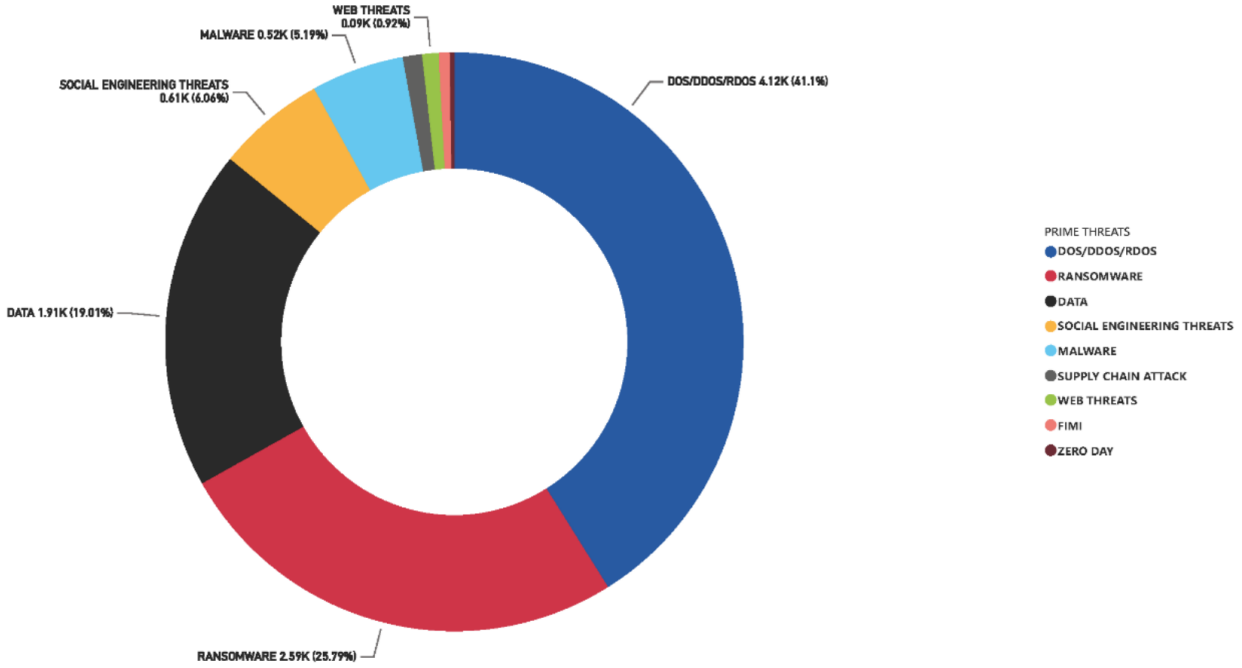


Рис. 1.1. Розподіл проаналізованих інцидентів за типом загрози (липень 2023 р. – червень 2024 р.) [1]

У Звіті Verizon [2] зазначається, що з року в рік з'являються нові та інноваційні атаки, а також варіації атак, які все ще залишаються успішними. Змінний ландшафт кіберзагроз може бути заплутаним і приголомшливим. Коли на додаток до різних типів атак додаються такі фактори, як людський фактор і/або погано захищені паролі, все стає ще більш заплутаним.

На рисунку 1.2 представлено найпопулярніші варіанти дій у порушеннях.

На рисунку 1.3 представлено найпопулярніші варіанти дій для інцидентів. DoS-атаки займають перше місце, оскільки вони присутні в 59% зареєстрованих інцидентів.

На рисунку 1.4 показано вектори дій у порушеннях. Спостерігалося значне зростання недбалості через збільшення помилкових порушень і зростання використання електронної пошти як вектора, викликаного збільшенням претексту. Однак веб-додатки експлуатуються поруч з використанням викрадених облікових даних і використанням вразливостей для проникнення у захист організацій [2].


Рис. 1.2. Найпопулярніші різновиди дій у порушеннях (n=9982) [2]


Рис. 1.3. Основні види дій в інцидентах (n=28 625) [2]

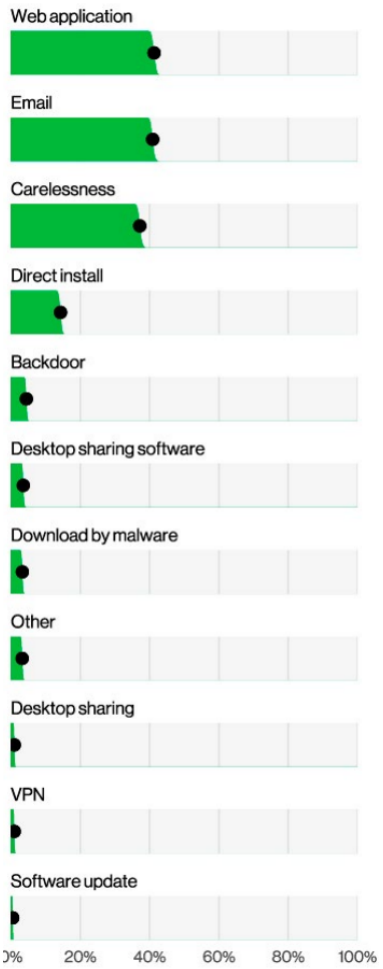


Рис. 1.4. Основні вектори дій у порушеннях (n=7248) [2]

Отже, проблема виявлення та реагування на інциденти безпеки в інформаційних системах організацій набула критичної актуальності в сучасному цифровому середовищі. Зростання кількості та складності кіберзагроз становить значний ризик для безперервності бізнес-процесів, збереження конфіденційної інформації та підтримання довіри клієнтів і партнерів [3]. Організації, незалежно від їхнього розміру та галузі діяльності, постійно стикаються зі спробами несанкціонованого доступу, шкідливим програмним забезпеченням, фішинговими атаками та іншими видами кіберзлочинів, які можуть призвести до значних фінансових втрат, репутаційних збитків та порушення нормального функціонування [4].

В умовах, коли кіберзагрози стають дедалі витонченішими та здатними обходити традиційні засоби захисту, ефективна система виявлення та реагування

на інциденти безпеки є невід’ємною частиною загальної стратегії кібербезпеки організації [3]. Недостатня підготовка та повільне реагування на інциденти можуть призвести до катастрофічних наслідків, тоді як своєчасне виявлення та адекватні заходи з локалізації та усунення загрози здатні мінімізувати потенційну шкоду.

## 1.2. Аналіз підходів до виявлення інцидентів безпеки в інформаційних системах організації та реагування на них

В контексті інформаційної безпеки, *інцидент безпеки визначається як подія або серія подій, що фактично або потенційно загрожують конфіденційності, цілісності або доступності інформаційної системи організації чи інформації, яку ця система обробляє, зберігає або передає* [4]. Інцидент може також становити порушення або неминучу загрозу порушення політик безпеки, процедур безпеки або правил прийнятного використання [5].

Інциденти безпеки можуть бути класифіковані за різними критеріями. За типом загрози виділяють інциденти, пов'язані зі шкідливим програмним забезпеченням (malware), фішингом, атаками типу «відмова в обслуговуванні» (DoS), витоками даних, несанкціонованим доступом тощо. За рівнем впливу інциденти можуть бути незначними, середніми або критичними, залежно від того, наскільки вони порушують бізнес-процеси та завдають шкоди організації. За джерелом інциденти поділяються на зовнішні (наприклад, кібератаки ззовні) та внутрішні (наприклад, дії недобросовісних співробітників або випадкові помилки персоналу) [6].

Важливо розрізняти інцидент безпеки, подію безпеки та вразливість. Подія безпеки є будь-якою значущою подією в інформаційній системі, яка може свідчити про можливе порушення безпеки або відхилення від нормального функціонування, але не обов’язково несе загрозу. Вразливість, у свою чергу, є слабким місцем в інформаційній системі, яке може бути використане зловмисником для реалізації загрози. Інцидент безпеки відбувається тоді, коли

вразливість була успішно використана або була здійснена спроба її використання, що призвело до порушення або потенційної загрози безпеці інформаційної системи чи даних. Розуміння цієї різниці є ключовим для ефективного управління інформаційною безпекою організації [7].

Процес виявлення інцидентів безпеки є критично важливим етапом у загальному циклі реагування на інциденти. Ефективне виявлення дозволяє організації своєчасно ідентифікувати потенційні загрози та вжити необхідних заходів для мінімізації їхнього впливу. Різні фреймворки, такі як NIST (Національний інститут стандартів і технологій США) та SANS (SysAdmin, Audit, Network, and Security Institute), пропонують структуровані підходи до процесу виявлення інцидентів безпеки.

NIST запропонував нову модель життєвого циклу реагування на інциденти (рисунок 1.5). Нижній рівень відображає те, що підготовчі заходи «Управління», «Виявлення» та «Захист» не є частиною самого реагування на інцидент. Скоріше, це набагато ширша діяльність з управління ризиками кібербезпеки, яка також підтримує реагування на інциденти. Реагування на інцидент показано на верхньому рівні малюнка: виявлення, реагування та відновлення. Крім того, потреба в безперервному вдосконаленні позначена середнім рівнем із категорією покращення в межах функції ідентифікації та пунктирними зеленими лініями. Уроки, отримані під час виконання всіх дій у всіх функціях, вводяться в покращення, і ці уроки аналізуються, розставляються за пріоритетами та використовуються для інформування всіх функцій [8].

Структура реагування на інциденти SANS [9] – це структурований підхід, який використовується організаціями для ефективного управління та пом’якшення інцидентів кібербезпеки. Він містить дієві кроки та вказівки, які допомагають службам реагування швидко виявляти, аналізувати та стримувати загрози безпеці. Ця структура забезпечує ретельне та організоване реагування на кіберінциденти, зменшуючи потенційний вплив на бізнес-операції та інфраструктуру безпеки.

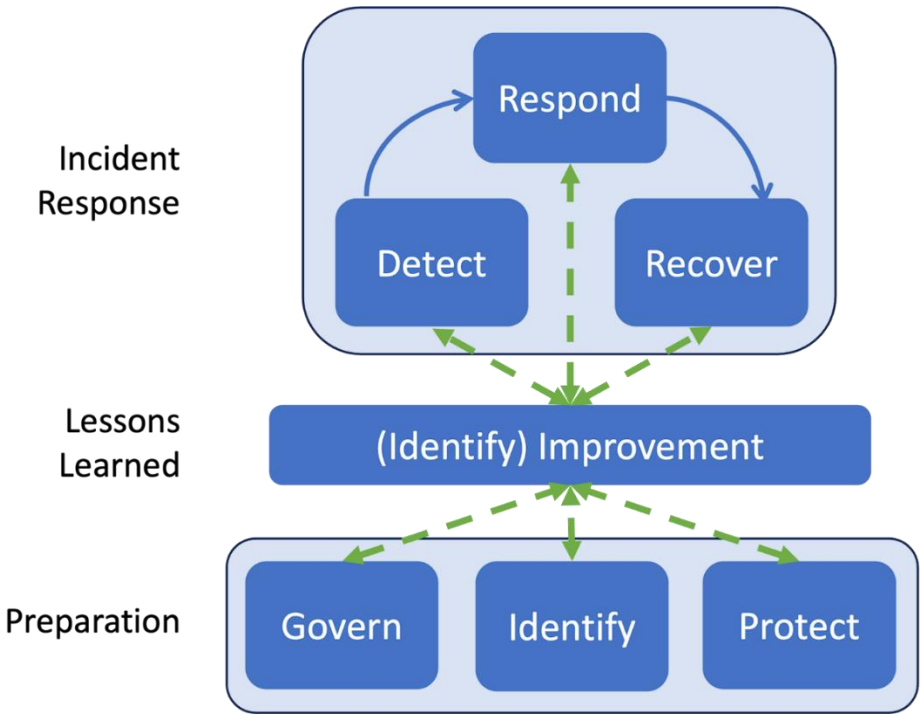


Рис. 1.5. Нова модель життєвого циклу реагування на інциденти за NIST [8]

Процес реагування на інцидент SANS включає наступні кроки: підготовка, ідентифікація, стримування, викорінення, відновлення та отримані уроки [9] (рисунок 1.6).


Рис. 1.6. Модель процесу реагування на інцидент за SANS [9]

Перший крок у процесі реагування на інциденти SANS *Підготовка* включає в себе налаштування правильних інструментів, політик і процедур напередодні будь-якого інциденту, гарантуючи, що команда реагування може діяти швидко та ефективно. Цей етап включає навчання персоналу, створення протоколів реагування на інциденти та встановлення стратегій комунікації як усередині, так і ззовні [9].

На цьому етапі організації також повинні зосередитися на надійній документації. Це означає збір і зберігання надійної контактної інформації для ключового персоналу, чітке визначення ролей і обов'язків і забезпечення доступу всіх членів команди до необхідних ресурсів. Гарна підготовка зменшує хаос і забезпечує більш контрольоване управління інцидентами.

*Ідентифікація* має вирішальне значення для розпізнавання ознак кіберінциденту. Він передбачає моніторинг і аналіз даних для виявлення будь-яких аномалій, які вказують на порушення безпеки. Для ефективної ідентифікації потрібні передові системи безпеки та кваліфікована команда, яка здатна швидко розпізнавати потенційні загрози в нешкідливих подіях. Виявивши інцидент на ранній стадії, організації можуть мінімізувати потенційну завдану шкоду [9].

Щоб зміцнити цей етап, компанії повинні інвестувати в постійне навчання та складні технології безпеки. Оновлення можливостей виявлення означає, що служби реагування можуть точно виявляти та посилювати інциденти, забезпечуючи відповідну терміновість і точність.

*Стримування* спрямоване на обмеження поширення та впливу інциденту. Негайні та довгострокові стратегії застосовуються, щоб запобігти подальшій шкоді, зберігаючи безперервність бізнесу. Цей крок може передбачати розділення мережі, блокування зловмисного трафіку або вимкнення скомпрометованих облікових записів. Ефективне стримування економить ресурси та дає більше часу для планування та виконання стратегічного процесу ліквідації [9].

Розробляючи стратегію стримування, важливо знайти баланс між сильними захисними діями та підтриманням основних бізнес-функцій. Швидке прийняття рішень, підкріплене чіткими заздалегідь визначеними планами стримування,

дозволяє негайно стабілізувати ситуацію.

*Ерадикація*: після стримування етап ліквідації включає повне усунення загрози з навколишнього середовища. Це може означати видалення шкідливих файлів, видалення скомпрометованого програмного забезпечення або усунення вразливостей, які були використані. Ретельне розслідування визначає першопричину інциденту, щоб ефективно усунути всі аспекти порушення [9].

Найкращі практики викорінення вимагають точної координації та документування, щоб гарантувати, що жодних залишків загрози не залишиться. Цей крок відновлює цілісність систем і готує їх до безпечного відновлення, зменшуючи ризик повторення.

На етапі *Відновлення* системи відновлюються та повертаються до нормального функціонування, гарантуючи, що вони вільні від будь-яких лазівок у безпеці. Цей етап включає ретельне тестування систем для перевірки безпеки та проведення комплексного моніторингу системи для підтвердження нормальної функціональності. Дії відновлення мають узгоджуватися з бізнес-пріоритетами, а системи мають бути повернуті в режим роботи контрольованим способом [9].

Плани відновлення мають бути розроблені таким чином, щоб зменшити майбутні ризики, враховуючи уроки, отримані з інциденту. Ця активна адаптація зміцнить системи проти майбутніх викликів безпеці.

*Отримані уроки*: цей останній крок передбачає аналіз інциденту від початку до кінця, щоб визначити успіхи та недоліки. Отримані уроки документуються та використовуються для посилення плану реагування на інциденти , зменшуючи ймовірність подібних інцидентів у майбутньому. Ретельний аналіз сприяє безперервному вдосконаленню та коригуванню стратегій, які необхідні для адаптації до мінливого ландшафту кіберзагроз [9].

Залучення всіх зацікавлених сторін до процесу перевірки забезпечує повне розуміння інциденту та сприяє розвитку культури постійного вдосконалення безпеки в організації.

*Одним з основних каналів є системи керування інформацією та подіями безпеки (SIEM). SIEM-системи здійснюють збір та агрегацію даних журналів з*

*різних джерел, таких як мережеві пристрої, сервери, кінцеві точки та програми [4]*. Вони виконують кореляцію подій, виявляючи закономірності та аномалії, які можуть свідчити про інцидент безпеки. SIEM-системи також часто включають функціональність аналізу поведінки користувачів та сутностей (UEBA), що дозволяє виявляти підозрілу активність на основі відхилень від встановлених базових ліній поведінки.

*Важливу роль у виявленні інцидентів відіграють системи виявлення вторгнень (IDS) та запобігання вторгненням (IPS).* IDS здійснюють моніторинг мережевого трафіку та системної активності на предмет відомих сигнатур атак або підозрілої поведінки, генеруючи попередження при їх виявленні. IPS, на відміну від IDS, здатні не лише виявляти, але й активно блокувати або запобігати вторгненням, наприклад, шляхом блокування шкідливого трафіку або розриву з'єднань. Методи виявлення, що використовуються в IDS/IPS, включають сигнатурний аналіз (пошук відомих шаблонів атак) та аналіз аномалій (виявлення відхилень від нормальної поведінки) [10].

*Аналіз журналів (Log Analysis)* є ще одним важливим методом виявлення інцидентів безпеки. Ретельний перегляд та аналіз журналів подій з різних систем та програм може виявити підозрілу активність, несанкціонований доступ або інші ознаки інцидентів безпеки. Аналіз журналів може включати виявлення аномалій, пошук індикаторів компрометації, а також аналіз першопричин інцидентів.

*Поведінкова аналітика (Behavioral Analytics)* використовує методи машинного навчання та штучного інтелекту для аналізу закономірностей поведінки користувачів та сутностей в інформаційній системі. Шляхом встановлення базових ліній нормальної поведінки, поведінкова аналітика може виявляти аномалії, які можуть свідчити про інсайдерські загрози, компрометацію облікових записів або інші складні атаки.

*Розширене виявлення та реагування (XDR)* є відносно новою категорією технологій, яка інтегрує дані безпеки з різних рівнів інфраструктури, включаючи кінцеві точки, мережу та хмарні середовища, для забезпечення більш цілісного та ефективного виявлення та реагування на загрози. XDR-системи здійснюють

кореляцію даних з різних джерел, що дозволяє виявляти складні атаки, які можуть бути непомітними при аналізі окремих компонентів інфраструктури.

Після виявлення потенційного інциденту безпеки важливим етапом є його верифікація. Не всі виявлені події є фактичними інцидентами безпеки, тому процес верифікації має на меті підтвердити, чи є виявлена подія справжнім інцидентом, чи це хибно-позитивне спрацьовування.

## 1.3. Аналіз існуючих рішень для виявлення та реагування на інциденти безпеки в інформаційній системі організації

Системи керування інформацією та подіями безпеки (SIEM) відіграють вирішальну роль у сучасній кібербезпеці для організацій будь-якого розміру. SIEM-системи агрегують та аналізують дані з різноманітних інструментів безпеки та IT. Вони перетворюють події та журнали на корисну інформацію для виявлення загроз, реагування на інциденти та забезпечення відповідності нормативним вимогам. Зважаючи на зростаючу складність кібератак та посилення вимог регуляторних органів, впровадження SIEM-рішень стає необхідністю.

Ми визначили та провели аналіз трьох провідних open-source SIEM-рішень, з акцентом на їхніх можливостях виявлення та реагування на інциденти в інформаційних системах організацій. Аналіз спирався на доступні дослідницькі матеріали для надання порівняльного огляду, що допоможе у виборі відповідного open-source SIEM.

SIEM агрегує дані з сотень інструментів безпеки та IT-інструментів на підприємстві та використовує статистичну кореляцію та правила для перетворення подій і записів журналу в корисну інформацію. Команди безпеки можуть використовувати цю інформацію для виявлення загроз у реальному часі, проведення судових розслідувань інцидентів безпеки, організації реагування на інциденти та підготовки до перевірок відповідності.

Сьогодні SIEM є стандартним підходом до безпеки. Оскільки кількість кібератак продовжує зростати, а правила безпеки стають дедалі суворішими,

багато організацій застосовують SIEM. Регуляторні зміни, такі як PCI DSS і GDPR ЄС, зробили обов'язковим видалення системних і додаткових журналів подій з окремих серверів і безпечне їх зберігання для дослідження та реагування.

Використовуючи інструменти SIEM з відкритим кодом, організації можуть зменшити витрати на ліцензування програмного забезпечення та оцінити конкретні можливості, перш ніж збільшувати інвестиції в продукт. Рішення SIEM з відкритим кодом має основні функції, які відповідають потребам невеликих організацій, які тільки починають реєструвати та аналізувати інформацію про події безпеки [11].

Обмеження SIEM з відкритим кодом [11]:

програмне забезпечення SIEM з відкритим кодом може вимагати багато зусиль у міру зростання організації;

навіть якщо ми заощадимо на витратах на ліцензування, можуть виникнути витрати на поточне обслуговування;

багато рішень SIEM з відкритим кодом не мають таких критичних можливостей SIEM, як звітування, кореляція подій і віддалене керування збором журналів. Таким чином, фахівцям з кібербезпеки може знадобитися об'єднати SIEM з відкритим кодом з іншими інструментами;

розгортання ефективної SIEM з відкритим кодом зазвичай вимагає високого рівня знань і часу. SIEM з відкритим кодом зазвичай не мають можливостей зберігання та керування. Це питання, про яке слід знати, враховуючи величезний обсяг даних.

Корпоративні рішення SIEM пропонують розширені можливості керування для найпоширеніших випадків використання, включаючи конфігурацію та встановлення, конфігурацію кореляції, фільтри та попередньо створені візуалізації. За допомогою такого рішення ви можете відстежувати діяльність у великих центрах обробки даних і централізовано керувати та налаштовувати додатки, пов'язані з безпекою [11].

Можливо, найголовніше те, що лише корпоративні SIEM наразі пропонують можливості SIEM наступного покоління. Корпоративна SIEM наступного

покоління містить дві нові технології, які економлять час команд безпеки та значно покращують виявлення інцидентів і реагування на них [11]:

UEBA (Аналіз поведінки користувачів і об'єктів) – це подальша еволюція правил і кореляцій, які використовують штучний інтелект і машинне навчання для аналізу моделей поведінки користувачів і IT-систем, які можуть вказувати на загрози. Виявляйте аномалії високого ризику;

оркестрація безпеки, автоматизація та реагування (SOAR) – це функція, інтегрована в корпоративні системи, яка керує системами для автоматизації процесів реагування на інциденти, таких як пом'якшення шкідливого програмного забезпечення та атак з крадіжки даних.

Таблиця 1.1

Найпопулярніші інструменти SIEM з відкритим кодом

| SIEM з відкритим кодом | Варіанти розгортання | Основна функція | Обмеження |
|---|---|---|---|
| **ELK stack**<br>Набір із трьох відкритих продуктів: Elasticsearch, Logstash і Kibana. Використовуються ці три інструменти, щоб досягти видимості та аналізу IT-подій. | Віртуальні середовища, фізичне обладнання, приватні хмари, приватні зони в публічних хмарах, публічні хмари (Google, Azure, AWS тощо). | Ведення журналів та аналіз журналів<br>Обробка, фільтрація, кореляція та покращення зібраних даних журналу<br>Індексація та зберігання даних часових рядів | Аналіз журналу загального призначення – не розроблений як система SIEM.<br>Немає вбудованих звітів і сповіщень.<br>Немає вбудованих правил безпеки. |
| **Prelude**<br>Інтегрується з різними іншими інструментами з відкритим кодом. Відкрита версія однойменного комерційного інструменту. | Linux, OpenBSD, FreeBSD, NetBSD, Sun/Solaris, MacOSX, Tru64 та інші системи на основі UNIX. | Корелювати, фільтрувати, сповіщати<br>Можливості аналізу та візуалізації | Використовується для дослідження, оцінки та тестування в дуже невеликих середовищах.<br>За словами виробника, продуктивність версії з відкритим кодом Prelude значно нижча, ніж у комерційного продукту Prelude SIEM. |
| **Платформа OSSIM**<br>SIEM, включаючи збір подій, нормалізацію та кореляцію. | Локальні фізичні та віртуальні середовища. | Виявлення активів<br>Оцінка вразливості<br>Корелюйте події SIEM<br>Виявлення вторгнень<br>Моніторинг | Проблеми з продуктивністю в масштабі.<br>Дуже обмежені можливості керування журналами.<br>Можна розгорнути лише на одному сервері. |

| | | | |
|---|---|---|---|
| | | поведінки | Немає інтеграції з рішенням UEBA.<br>Обмежені можливості моніторингу програми та бази даних.<br>Обмежені можливості графічної бази даних і часткова власна аналітика користувача.<br>Немає підтримки та інтеграції інструментів DAM, CASB, DAP, DLP. |

ELK Stack – це потужна комбінація трьох інструментів з відкритим вихідним кодом: Elasticsearch, Logstash і Kibana, які разом забезпечують комплексне рішення для аналізу та моніторингу журналів (рисунок 1.7) [12].

Elasticsearch – це система розподіленого пошуку та аналітики, яка є основою стеку ELK. Він створений на базі Apache Lucene і призначений для зберігання, пошуку та отримання величезних обсягів даних у режимі реального часу. Elasticsearch гарно справляється з обробкою неструктурованих даних, що робить його ідеальним рішенням для керування журналами.

Ключові особливості Elasticsearch [12]:

RESTful API для легкої інтеграції з програмами;

індексація та пошук даних майже в реальному часі;

горизонтально масштабована для обробки великих наборів даних і великих навантажень;

підтримка повнотекстового пошуку, агрегацій і аналітики.

Logstash – це конвеєр обробки даних, який приймає та перетворює дані журналу з різних джерел перед тим, як відправити їх до Elasticsearch для індексування. Він діє як централізований агент збору даних, підтримуючи широкий спектр джерел введення, включаючи журнали, показники та інші дані про події. Наприклад, у нас може бути тема *kafka*, у якій ми надсилатимемо дані журналу за допомогою *filebeat*, а за допомогою *logstash* ми зможемо зберігати всі дані журналу для подальшого аналізу.

Основні характеристики Logstash:

розширені модулі введення для різних джерел даних;

потужні плагіни фільтрів для перетворення та збагачення даних;

плагіни виводу для надсилання даних до Elasticsearch або інших місць призначення.

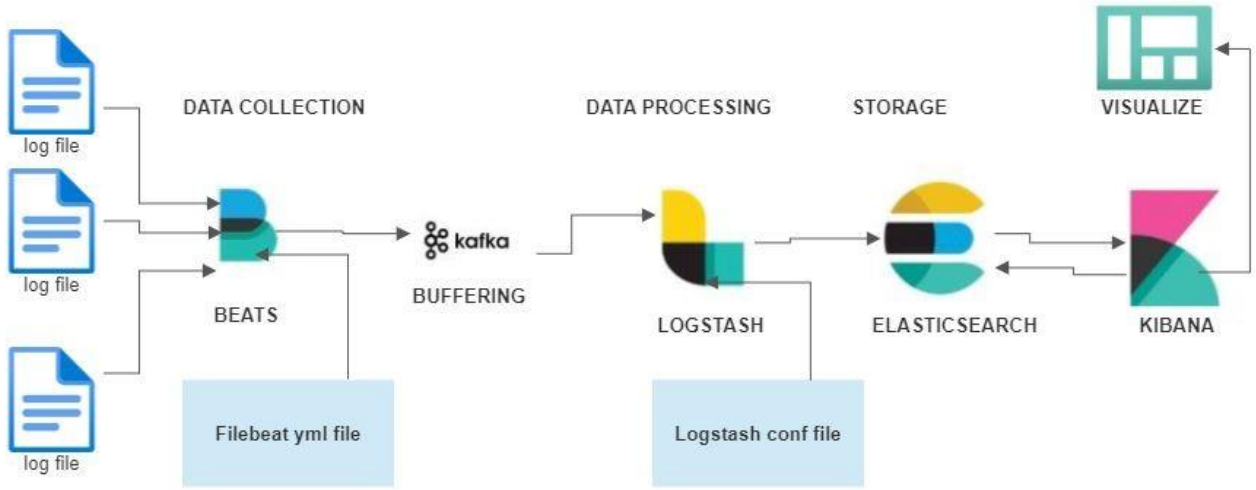


Рис. 1.7. Архітектура ELK stack [12]

Kibana – це веб-інструмент візуалізації даних, який забезпечує зручний інтерфейс для дослідження, аналізу та візуалізації даних, що зберігаються в Elasticsearch. Це дозволяє користувачам створювати інтерактивні інформаційні панелі, діаграми та графіки, щоб отримати цінну інформацію з даних журналу.

Основні характеристики Kibana [12]:

настроювані інформаційні панелі з візуалізаціями за допомогою перетягування;

багаті можливості агрегації для узагальнення даних;

дослідження та фільтрація даних у реальному часі;

інтеграція з Elasticsearch Query DSL для складних запитів.

Випадки використання стека ELK [12]:

аналіз журналів: стек ELK широко використовується для аналізу журналів у різних галузях промисловості. Це дозволяє адміністраторам і розробникам ефективно відстежувати та усувати неполадки системи, візуалізуючи дані журналу в режимі реального часу;

моніторинг безпеки: ELK можна використовувати для моніторингу безпеки,

виявлення та аналізу подій безпеки та виявлення потенційних загроз;

бізнес-аналітика: можливості візуалізації даних ELK роблять його придатним для бізнес-аналітики та аналізу даних, надаючи інформацію з різних джерел даних;

моніторинг продуктивності додатків (APM): ELK можна інтегрувати з інструментами APM для відстеження продуктивності додатків і виявлення вузьких місць.

Отже, ELK Stack є надійним та універсальним рішенням для аналізу та моніторингу журналів, яке широко поширене завдяки своїй масштабованості, можливостям роботи в реальному часі та відкритому коду. Завдяки потужним можливостям індексування та пошуку Elasticsearch, гнучкій обробці даних Logstash та інтуїтивно зрозумілій візуалізації Kibana ELK дозволяє організаціям використовувати потенціал своїх журнальних даних і отримувати цінну інформацію для кращого прийняття рішень і усунення вразливостей.

AT&T Open Source Security Information Management (OSSIM) – це версія з відкритим кодом платформи AlienVault USM Anywhere SIEM. Це дозволяє групам безпеки збирати та аналізувати дані про події безпеки з багатьох різних джерел і допомагає їм виконувати завдання з керування вразливістю та моніторингу поведінки [13].

OSSIM має більш обмежений набір функцій, ніж USM Anywhere, що може зробити його менш ефективним у певних середовищах. Наприклад, OSSIM можна розгорнути лише на одному сервері та не захищає хмарні служби. Ручне керування плагінами може бути трудомістким завданням, яке USM Anywhere значно спрощує.

Prelude – це структура SIEM, яка об’єднує багато інших інструментів в єдиний інтерфейс. Подібно до інших платформ з відкритим кодом у цьому списку, Prelude доступний як безкоштовна платформа з відкритим кодом з обмеженою функціональністю, так і платна опція, готова для корпоративних клієнтів.

Варіант Prelude з відкритим кодом має значні обмеження, які роблять його

неможливим для активної комерційної організації. Сама компанія пропонує версію з відкритим вихідним кодом із застереженням про те, що вона розроблена для «цілей оцінювання, дослідження та тестування в дуже невеликих середовищах», зі значно нижчою продуктивністю, ніж платна версія того самого продукту [13].

SIEM з відкритим кодом значно зросли за останні кілька десятиліть і успішно розгортаються багатьма організаціями. Однак загальновідомо, що вартість ліцензування становить лише невелику частку загальної вартості власника системи SIEM, навіть незважаючи на те, що основним рушієм впровадження стало зниження витрат на ліцензування. Інші компоненти, які можуть бути більшими за саму SIEM, включають [11]:

обладнання та сховище можуть бути дорогими та складними в управлінні, особливо для середнього та великого бізнесу;

час аналітика є найціннішим ресурсом для більшості команд безпеки, і він необхідний для будь-якої форми оповіщення SIEM.

# 2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ІНЦИДЕНТИ БЕЗПЕКИ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА БАЗІ WAZUH

## 2.1. Призначення та основні функції рішення Wazuh

Платформа Wazuh надає функції XDR і SIEM для захисту робочих навантажень хмари, контейнера та сервера. Вони включають аналіз даних журналу, виявлення вторгнень і шкідливого програмного забезпечення, моніторинг цілісності файлів, оцінку конфігурації, виявлення вразливостей і підтримку нормативної відповідності [14].

*Рішення Wazuh базується на агенті Wazuh, який розгортається на контрольованих кінцевих точках, і на трьох центральних компонентах: сервері Wazuh, індексаторі Wazuh і інформаційній панелі Wazuh [14].*

*Індексатор Wazuh* – це високомасштабована система повнотекстового пошуку та аналітики. Цей центральний компонент індексує та зберігає сповіщення, створені сервером Wazuh.

*Сервер Wazuh* аналізує дані, отримані від агентів. Він обробляє його 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.

*Інформаційна панель Wazuh* – це веб-інтерфейс користувача для візуалізації та аналізу даних. Він включає в себе готові інформаційні панелі для пошуку загроз, відповідності нормативним вимогам (наприклад, PCI DSS, GDPR, CIS, HIPAA, NIST 800-53), виявлені вразливі програми, дані моніторингу цілісності файлів, результати оцінки конфігурації, події моніторингу хмарної інфраструктури та інші. Він 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 [14].

На рисунку 2.1 показано компоненти та потік даних у Wazuh.

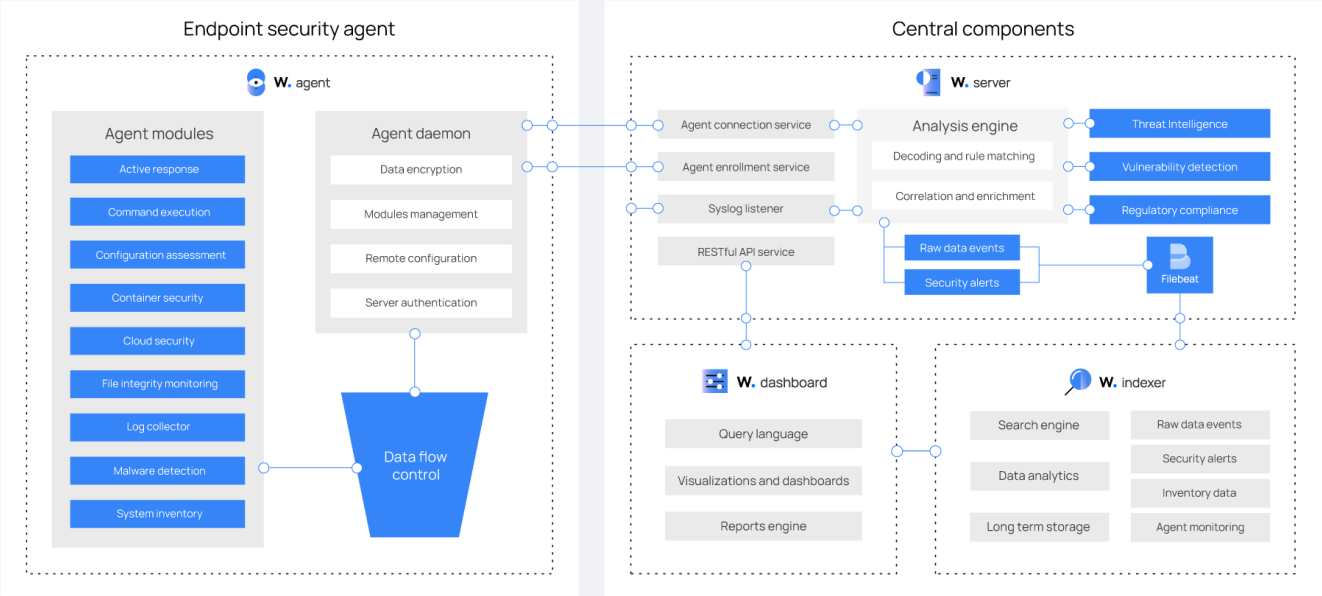


Рис. 2.1. Компоненти та потік даних у Wazuh [14]

Індексатор Wazuh – це високомасштабована система повнотекстового пошуку та аналітики. Цей центральний компонент Wazuh індексує та зберігає сповіщення, створені сервером Wazuh, і забезпечує пошук даних і аналітику майже в реальному часі. Індексатор Wazuh можна налаштувати як однохостовий або багатохостовий кластер, що забезпечує масштабованість і високу доступність.

Індексатор Wazuh зберігає дані як документи JSON. Кожен документ

співвідносить набір ключів, імен полів або властивостей із відповідними значеннями, які можуть бути рядками, числами, логічними значеннями, датами, масивами значень, геолокаціями чи іншими типами даних [14].

Індексатор Wazuh – це сукупність пов'язаних між собою документів. Документи, що зберігаються в індексаторі Wazuh, розподіляються між різними контейнерами, відомими як сегменти. Розподіляючи документи між декількома шарами та розподіляючи ці сегменти між кількома вузлами, індексатор Wazuh може забезпечити надлишковість. Це захищає вашу систему від апаратних збоїв і збільшує пропускну здатність запитів, коли вузли додаються до кластера.

Wazuh використовує чотири різні індекси для зберігання різних типів подій (таблиця 2.1).

Таблиця 2.1

Індекси для зберігання різних типів подій в Wazuh [14]

| Індекс | опис |
|---|---|
| wazuh – **сповіщення** | Зберігає сповіщення, створені сервером Wazuh . Вони створюються кожного разу, коли подія запускає правило з достатньо високим пріоритетом (цей поріг можна налаштувати). |
| wazuh - **архіви** | Зберігає всі події (архівні дані), отримані сервером Wazuh , незалежно від того, чи спрацьовує правило. |
| wazuh - **моніторинг** | Зберігає дані, пов'язані зі статусом агента Wazuh протягом певного часу. Він використовується веб-інтерфейсом для відображення того, коли окремі агенти є або були `Active` , `Disconnected` , або . `Never connected` |
| wazuh - **статистика** | Зберігає дані, пов'язані з продуктивністю сервера Wazuh . Він використовується веб-інтерфейсом для представлення статистики продуктивності. |

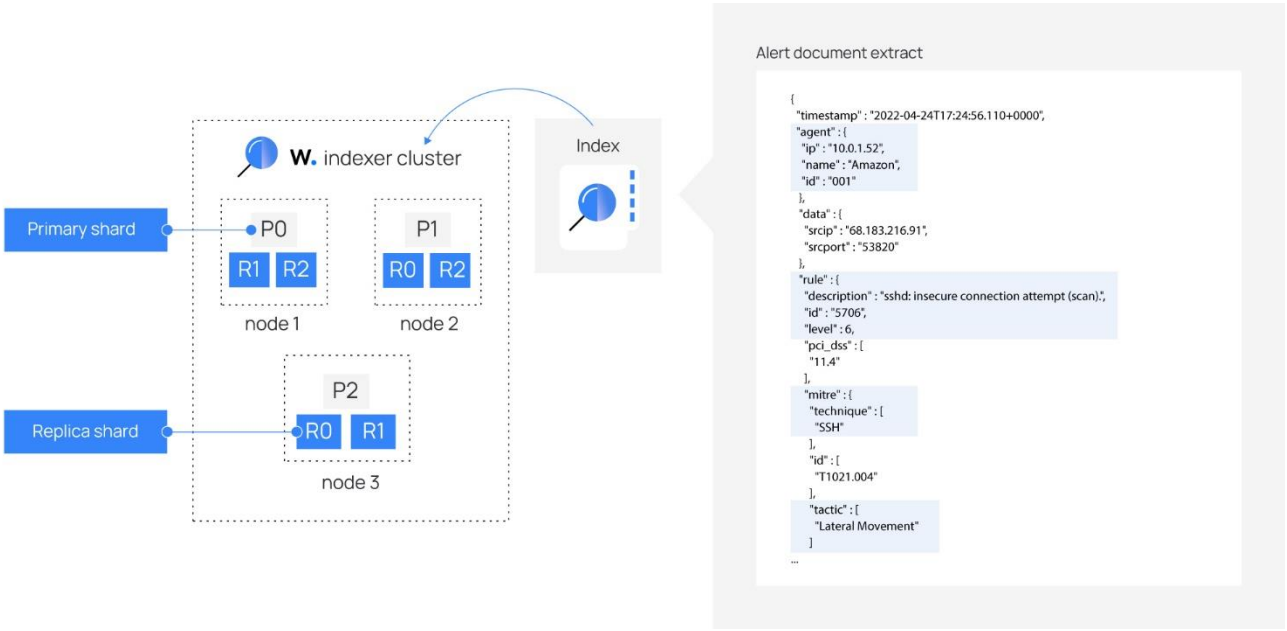


Рис. 2.2. Схема індексування в Wazuh [14]

Ми можемо взаємодіяти з кластером індексатора Wazuh за допомогою REST API індексатора Wazuh, який пропонує велику гнучкість. Ми можемо виконувати пошук, додавати або видаляти документи, змінювати індекси тощо.

Ось приклад запиту до індексатора Wazuh, який повертає сповіщення про останнє бокове переміщення за допомогою техніки SSH [14]:

```
GET /wazuh-alerts-4.x-*/_search
{
  "query": {
    "bool": {
      "must": [
        {"term": { "rule.mitre.tactic": "Lateral Movement" } },
        {"term": { "rule.mitre.technique": "SSH" } }
      ]
    }
  },
  "sort": [
    { "timestamp": { "order": "desc" } }
  ],
  "size": 1
}
```

Нижче наведено фрагмент результату запиту, який є частиною індексованого документа сповіщення:

```
{
   "timestamp" : "2022-04-24T17:24:56.110+0000",
   "agent" : {
    "ip" : "10.0.1.52",
    "name" : "Amazon",
    "id" : "001"
   },
   "data" : {
     "srcip" : "68.183.216.91",
     "srcport" : "53820"
   },
   "rule" : {
     "description" : "sshd: insecure connection attempt (scan).",
     "id" : "5706",
     "level" : 6,
     "pci_dss" : ["11.4"],
     "mitre" : {
       "technique" : [
         "SSH"
       ],
       "id" : ["T1021.004"],
       "tactic" : [
         "Lateral Movement"
      ]
     }
   },
   "full_log" : "Apr 24 17:24:55 ip-10-0-1-52 sshd[32179]: Did not receive identification string from 6
   "location" : "/var/log/secure",
   "predecoder" : {
     "hostname" : "ip-10-0-1-52",
     "program_name" : "sshd",
     "timestamp" : "Apr 24 17:24:55"
   },
   "decoder" : {
     "parent" : "sshd",
     "name" : "sshd"
   },
   "GeoLocation" : {
     "city_name" : "Frankfurt am Main",
     "country_name" : "Germany",
     "region_name" : "Hesse"
   }
}
```


Індексатор Wazuh добре підходить для чутливих до часу випадків використання, таких як аналітика безпеки та моніторинг інфраструктури, оскільки це пошукова платформа майже в реальному часі. Затримка від часу індексації документа до моменту, коли він стає доступним для пошуку, дуже мала, зазвичай одна секунда [14].

Окрім швидкості, масштабованості та стійкості, індексатор Wazuh має

кілька вбудованих функцій, які роблять зберігання та пошук даних ще ефективнішим, наприклад зведення даних, сповіщення, виявлення аномалій та керування життєвим циклом індексу.

Серверний компонент Wazuh аналізує дані, отримані від агентів , ініціюючи сповіщення при виявленні загроз або аномалій. Він також використовується для віддаленого керування конфігурацією агентів і моніторингу їх стану [14].

Сервер Wazuh використовує джерела аналізу загроз, щоб покращити свої можливості виявлення. Він також збагачує дані попереджень, використовуючи структуру MITER ATT&CK і нормативні вимоги, такі як PCI DSS, GDPR, HIPAA, CIS і NIST 800-53, надаючи корисний контекст для аналітики безпеки.

Крім того, сервер Wazuh можна інтегрувати із зовнішнім програмним забезпеченням, включаючи системи продажу квитків, такі як ServiceNow, Jira та PagerDuty, а також платформи обміну миттєвими повідомленнями, такі як Slack. Ці інтеграції зручні для оптимізації операцій безпеки [14].

Сервер Wazuh запускає механізм аналізу, Wazuh RESTful API, службу реєстрації агентів, службу підключення агентів, демон кластера Wazuh і Filebeat. Сервер інстальовано в операційній системі Linux і зазвичай працює на автономній фізичній машині, віртуальній машині, контейнері докерів або хмарному екземплярі.

На рисунку 2.3 показано архітектуру та компоненти сервера Wazuh.

Сервер Wazuh складається з кількох перелічених нижче компонентів, які мають різні функції, 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. Ця служба надає та розповсюджує унікальні ключі автентифікації кожному агенту. Процес працює як мережева служба та підтримує автентифікацію за допомогою сертифікатів TLS/SSL або шляхом надання фіксованого пароля.

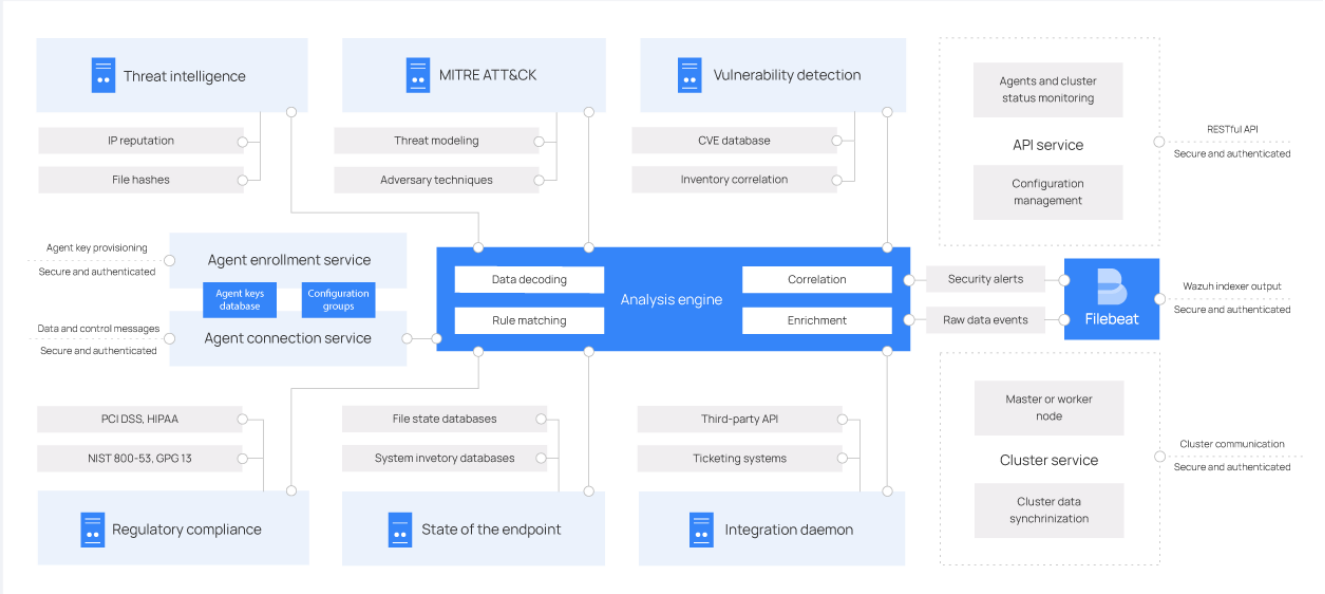


Рис. 2.3. Архітектура та компоненти сервера Wazuh [14]

Служба підключення агентів: ця служба отримує дані від агентів. Він використовує ключі, надані службою реєстрації, для перевірки ідентифікації кожного агента та шифрування зв'язку між агентом Wazuh і сервером Wazuh. Крім того, ця послуга забезпечує централізоване керування конфігурацією, що дозволяє віддалено надсилати нові параметри агента [14].

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, заборона IP-адреси, зупинка запущеного процесу або видалення артефакту зловмисного програмного забезпечення).

Wazuh RESTful API: ця служба 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.

Демон кластера Wazuh: ця служба використовується для горизонтального масштабування серверів Wazuh, розгортаючи їх як кластер. Така конфігурація в поєднанні з балансувальником мережевого 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. Він також забезпечує балансування навантаження при підключенні до багатохостового кластера індексатора Wazuh.

Інформаційна панель Wazuh – це гнучкий та інтуїтивно зрозумілий веб-інтерфейс користувача для видобутку, аналізу та візуалізації подій безпеки та даних сповіщень. Він також використовується для управління та моніторингу платформи Wazuh. Крім того, він надає функції для керування доступом на основі ролей (RBAC) і єдиного входу (SSO).

Візуалізація та аналіз даних. Веб-інтерфейс допомагає користувачам переміщатися між різними типами даних, зібраних агентом Wazuh, а також сповіщеннями безпеки, створеними сервером Wazuh. Користувачі також можуть створювати звіти та створювати власні візуалізації та інформаційні панелі [9].

Як приклад, Wazuh надає готові інформаційні панелі для відповідності нормативним вимогам, таким як PCI DSS, GDPR, HIPAA та NIST 800-53. Він також надає інтерфейс для навігації між структурою MITRE ATT&CK і пов’язаними сповіщеннями.

Моніторинг та налаштування агентів. Інформаційна панель Wazuh дозволяє користувачам керувати конфігурацією агентів і відстежувати їхній статус. Наприклад, для кожної контрольованої кінцевої точки користувачі можуть визначати, які модулі агентів будуть увімкнені, які файли журналів читатимуться, які файли відстежуватимуться на предмет змін цілісності або які перевірки

конфігурації виконуватимуться [14].

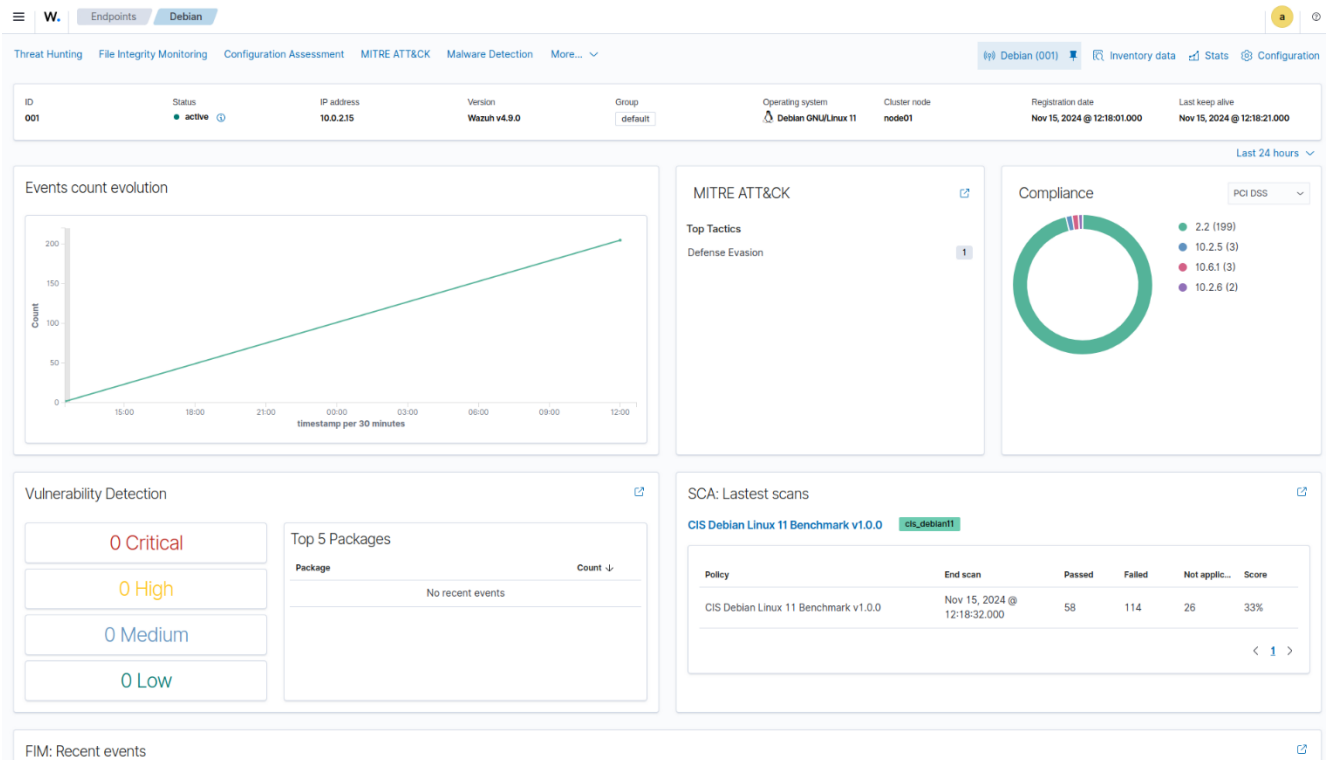


Рис. 2.3. Інформаційна панель Wazuh

Інформаційна панель Wazuh надає інтерфейс користувача, призначений для керування розгортанням Wazuh. Це включає моніторинг стану, журналів і статистики різних компонентів Wazuh. Він також включає налаштування сервера Wazuh і створення спеціальних правил і декодерів для аналізу журналів і виявлення загроз [14].

Інформаційна панель Wazuh містить інструмент перевірки набору правил, який може обробляти повідомлення журналу, щоб перевірити, як вони декодуються та чи відповідають вони правилу виявлення загроз чи ні. Ця функція особливо корисна, коли створено спеціальні декодери та правила, і користувач хоче їх протестувати.

Інформаційна панель Wazuh також містить консоль API для взаємодії користувачів з API Wazuh. Це можна використовувати для керування розгортанням Wazuh (наприклад, керування конфігураціями сервера чи агента, моніторинг стану та повідомлень журналу, додавання чи видалення агентів тощо).

Агент Wazuh працює в Linux, Windows, macOS, Solaris, AIX та інших операційних системах. Його можна розгорнути на ноутбуках, настільних комп'ютерах, серверах, хмарних примірниках, контейнерах або віртуальних машинах. Агент допомагає захистити вашу систему, надаючи можливості запобігання загрозам, їх виявлення та реагування. Він також використовується для збору різних типів даних системи та програм, які він пересилає на сервер Wazuh через зашифрований та автентифікований канал.

Агент Wazuh має модульну архітектуру. Кожен компонент відповідає за свої власні завдання, включаючи моніторинг файлової системи, читання повідомлень журналу, збір даних інвентаризації, сканування конфігурації системи та пошук шкідливих програм. Користувачі можуть керувати модулями агентів за допомогою налаштувань конфігурації, адаптуючи рішення до своїх конкретних випадків використання.

На рисунку 2.4 показано архітектуру та компоненти агента.

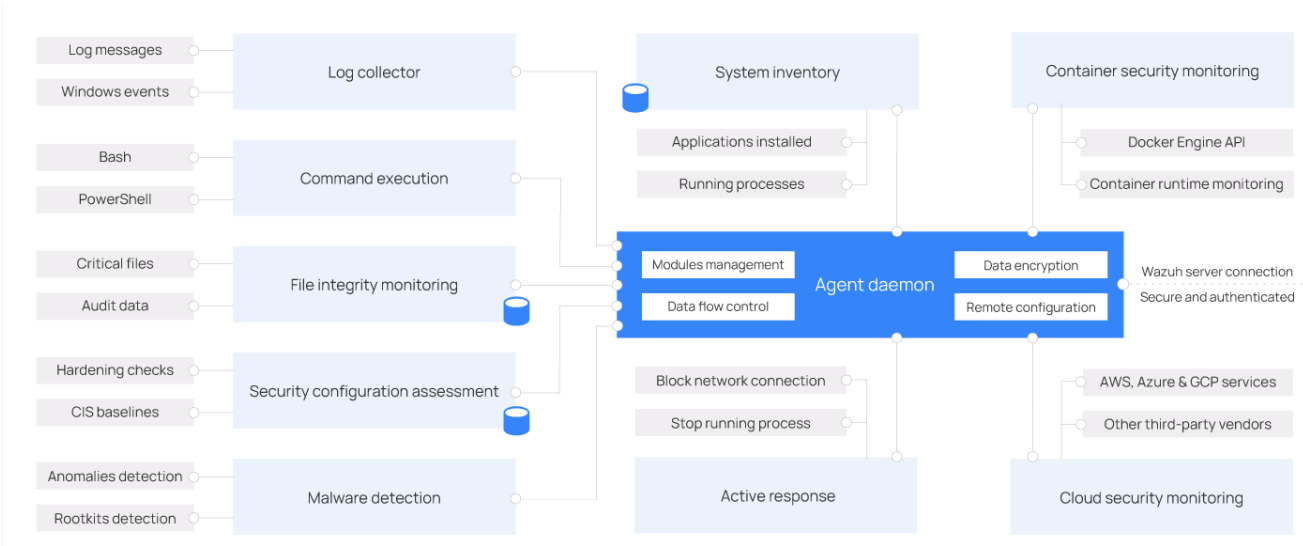


Рис. 2.4. Архітектура та компоненти агента Wazuh [14]

Усі модулі агентів можна налаштувати та виконувати різні завдання безпеки. Ця модульна архітектура дозволяє вмикати або вимикати кожен компонент відповідно до потреб безпеки. Нижче ви можете дізнатися про різні цілі всіх модулів агента.

Колектор логів: цей компонент агента може читати плоскі файли журналів і події Windows, збираючи повідомлення журналу операційної системи та програм. Він підтримує фільтри XPath для подій Windows і розпізнає багаторядкові формати, такі як журнали аудиту Linux. Він також може збагачувати події JSON додатковими метаданими.

Виконання команд: агенти періодично запускають авторизовані команди, збирають їхні результати та повідомляють про них на сервер Wazuh для подальшого аналізу. Ми можемо використовувати цей модуль для різних цілей, наприклад для моніторингу вільного місця на жорсткому диску або отримання списку останніх користувачів, які входили в систему.

Моніторинг цілісності файлів (File integrity monitoring, FIM): цей модуль відстежує файлову систему, повідомляючи про створення, видалення або зміну файлів. Він відстежує зміни в атрибутах файлів, дозволах, власності та вмісті. Коли відбувається подія, він фіксує подробиці про те, хто, що та коли в реальному часі. Крім того, модуль FIM створює та підтримує базу даних із станом відстежуваних файлів, що дозволяє виконувати запити віддалено.

Оцінка конфігурації безпеки (Security configuration assessment, SCA): цей компонент забезпечує безперервну оцінку конфігурації, використовуючи стандартні перевірки на основі тестів Центру безпеки в Інтернеті (CIS). Користувачі також можуть створювати власні перевірки SCA, щоб контролювати та застосовувати свої політики безпеки.

Інвентаризація системи: цей модуль агента періодично виконує сканування, збираючи дані інвентаризації, такі як версія операційної системи, мережеві інтерфейси, запущені процеси, встановлені програми та список відкритих портів. Результати сканування зберігаються в локальних базах даних SQLite, які можна запитувати віддалено.

Виявлення зловмисного програмного забезпечення: використовуючи підхід без сигнатур, цей компонент здатний виявляти аномалії та можливу присутність руткітів. Крім того, він шукає приховані процеси, приховані файли та приховані порти під час моніторингу системних викликів.

Активна відповідь: цей модуль запускає автоматичні дії при виявленні загроз, ініціюючи відповіді на блокування мережевого з'єднання, зупинку запущеного процесу або видалення шкідливого файлу. Користувачі також можуть створювати власні відповіді, коли це необхідно, і налаштовувати, наприклад, відповіді для запуску двійкового файлу в пісочниці, захоплення мережевого трафіку та сканування файлу антивірусом.

Моніторинг безпеки контейнера: цей модуль агента інтегровано з API Docker Engine для моніторингу змін у контейнерному середовищі. Наприклад, він виявляє зміни в зображеннях контейнерів, конфігурації мережі або обсягах даних. Крім того, він сповіщає про контейнери, що працюють у привілейованому режимі, і про користувачів, які виконують команди у запущеному контейнері.

Моніторинг хмарної безпеки: цей компонент відстежує хмарні постачальники, такі як Amazon Web Services, Microsoft Azure або Google GCP. Він нативно спілкується з їхніми API. Він здатний виявляти зміни в хмарній інфраструктурі (наприклад, створюється новий користувач, змінюється група безпеки, зупиняється хмарний екземпляр тощо) і збирати дані журналу хмарних служб (наприклад, AWS Cloudtrail, AWS Macie, AWS GuardDuty, Azure Active Directory тощо)

Агент Wazuh зв'язується з сервером Wazuh для передачі зібраних даних і подій, пов'язаних із безпекою. Крім того, агент надсилає оперативні дані, повідомляючи про свою конфігурацію та статус. Після підключення агент можна оновлювати, контролювати та налаштовувати віддалено з сервера Wazuh.

Зв'язок агента з сервером відбувається через захищений канал (TCP або UDP), що забезпечує шифрування та стиснення даних у реальному часі. Крім того, він містить механізми керування потоком, щоб уникнути затоплення, постановки подій у чергу, коли це необхідно, і захист пропускної здатності мережі.

Нам потрібно зареєструвати агента перед першим підключенням до сервера. Цей процес надає агенту унікальний ключ, який використовується для автентифікації та шифрування даних.

## 2.2. Архітектура рішення 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 [14].

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. Багатохостові кластери рекомендуються, якщо є багато контрольованих кінцевих точок, коли очікується великий обсяг даних або коли потрібна висока доступність.

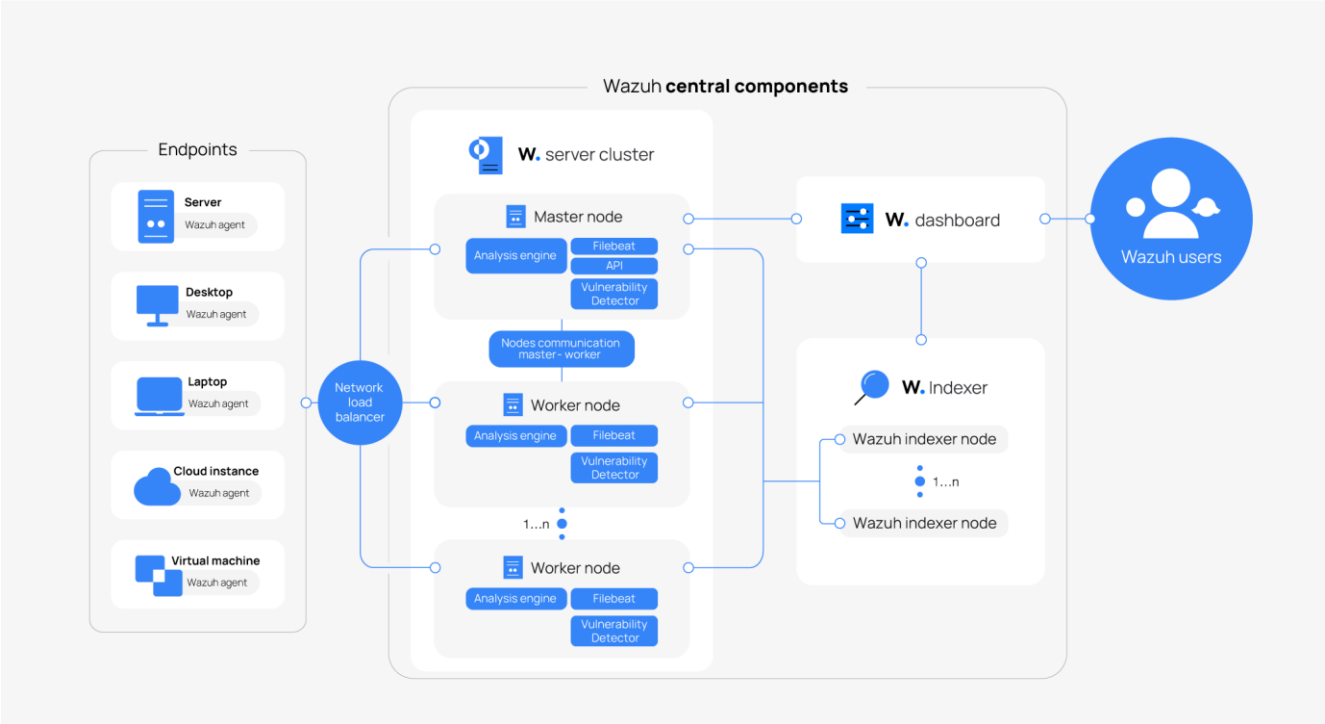


Рис. 2.5. Архітектура рішення Wazuh [14]

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 [14].

На рисунку 2.5 показано архітектуру розгортання Wazuh. Ми бачимо компоненти рішення та те, як сервер Wazuh і вузли індексатора Wazuh можуть бути налаштовані як кластери, забезпечуючи балансування навантаження та високу доступність.

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 (це можна налаштувати). 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 обидва файли, залежно від того, чи спрацьовує правило [14]:

файл */var/ossec/logs/archives/archives.json* містить усі події незалежно від того, чи спрацювало вони правило чи ні;

файл */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 із 128 бітами на блок і 256-бітними ключами. Шифрування Blowfish необов’язкове.

Розглянемо взаємодію сервера та індексатора Wazuh.

Сервер Wazuh використовує Filebeat для безпечної передачі даних попереджень і подій до індексатора Wazuh за допомогою шифрування TLS. Filebeat відстежує вихідні дані з сервера Wazuh і пересилає їх до індексатора Wazuh, який за замовчуванням слухає порт 9200/TCP. Після індексації ми можемо проаналізувати та візуалізувати дані на інформаційній панелі Wazuh.

Модуль виявлення вразливостей оновлює список вразливостей. Він також генерує сповіщення, надаючи інформацію про вразливі місця системи.

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. В таблиці 2.2 наведено список стандартних портів, які використовуються цими службами. За потреби користувачі можуть змінювати ці номери портів.

Таблиця 2.2

Список стандартних портів, які використовуються службами Wazuh [14]

| Component | Port | Protocol | Purpose |
|---|---|---|---|
| Wazuh server | 1514 | TCP (default) | Agent connection service |
| | 1514 | UDP (optional) | Agent connection service (disabled by default) |
| | 1515 | TCP | Agent enrollment service |
| | 1516 | TCP | Wazuh cluster daemon |
| | 514 | UDP (default) | Wazuh Syslog collector (disabled by default) |
| | 514 | TCP (optional) | Wazuh Syslog collector (disabled by default) |
| | 55000 | TCP | Wazuh server RESTful API |
| Wazuh indexer | 9200 | TCP | Wazuh indexer RESTful API |
| | 9300-9400 | TCP | Wazuh indexer cluster communication |
| Wazuh dashboard | 443 | TCP | Wazuh web user interface |

Архівне зберігання даних. 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*). Ці файли щодня стискаються та підписуються за допомогою контрольних сум MD5, SHA1 і SHA256. Структура каталогу та імені файлу така [9]:

```
root@wazuh-manager:/var/ossec/logs/archives/2022/Jan# ls -l
```

```
total 176
-rw-r----- 1 wazuh wazuh 234350 Jan  2 00:00 ossec-archive-01.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-01.json.sum
-rw-r----- 1 wazuh wazuh 176221 Jan  2 00:00 ossec-archive-01.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-01.log.sum
-rw-r----- 1 wazuh wazuh 224320 Jan  2 00:00 ossec-archive-02.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-02.json.sum
-rw-r----- 1 wazuh wazuh 151642 Jan  2 00:00 ossec-archive-02.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-02.log.sum
-rw-r----- 1 wazuh wazuh 315251 Jan  2 00:00 ossec-archive-03.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-03.json.sum
-rw-r----- 1 wazuh wazuh 156296 Jan  2 00:00 ossec-archive-03.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-03.log.sum
```

Рекомендується ротація та резервне копіювання архівних файлів відповідно до обсягу пам’яті сервера Wazuh. Використовуючи завдання cron, ми можемо легко зберігати лише певний часовий проміжок архівних файлів локально на сервері, наприклад, минулого року чи останніх трьох місяців.

З іншого боку, ми можемо відмовитися від зберігання архівних файлів і просто покладатися на індексатор Wazuh для зберігання архіву. Ця альтернатива може бути кращою, якщо ви періодично виконуєте резервне копіювання моментальних знімків індексатора Wazuh і/або маєте багатохостовий кластер індексатора Wazuh із копіями фрагментів для високої доступності. Ми навіть можемо використовувати завдання *cron*, щоб перемістити зроблені знімки індексів на кінцевий сервер зберігання даних і підписати їх за допомогою алгоритмів хешування MD5, SHA1 і SHA256.

## 2.3. Порядок функціонування рішення Wazuh

Рішення Wazuh Security Information and Event Management (SIEM) – це централізована платформа для агрегування й аналізу телеметрії в режимі реального часу для виявлення загроз і відповідності. Wazuh збирає дані про події з різних джерел, таких як кінцеві точки, мережеві пристрої, хмарні робочі навантаження та програми для ширшого охоплення безпеки [16].

Рішення Wazuh дозволяє отримувати сповіщення в режимі реального часу, коли виникають інциденти безпеки. Wazuh корелює події з багатьох джерел, інтегрує канали розвідки про загрози та надає настроювані інформаційні панелі та звіти. Ми можемо налаштувати сповіщення відповідно до конкретних вимог. Це дозволяє командам безпеки швидко реагувати на загрози та мінімізувати вплив інцидентів безпеки.

Рішення Wazuh дозволяє захистити корпоративну інфраструктуру та забезпечити дотримання нормативних вимог, відстежуючи та перевіряючи діяльність кінцевих точок. Wazuh збирає, зберігає та аналізує дані про події безпеки, щоб виявити аномалії або ознаки компрометації. Платформа SIEM додає контекстну інформацію до сповіщень, щоб прискорити розслідування та скоротити середній час відповіді [16].

Аналіз даних журналу – це важливий процес, який передбачає перевірку та отримання цінної інформації з файлів журналу, створених різними системами, програмами чи пристроями. Ці журнали містять записи подій, які надають корисну інформацію для усунення несправностей, аналізу та моніторингу безпеки та оптимізації продуктивності. Аналіз даних журналу є важливою практикою, яка сприяє безпечній, ефективній і надійній IT-екосистемі [14].

Wazuh збирає, аналізує та зберігає журнали з кінцевих точок, мережевих пристроїв і програм. Агент Wazuh, що працює на контрольованій кінцевій точці, збирає та пересилає журнали системи та програм на сервер Wazuh для аналізу. Крім того, ви можете надсилати повідомлення журналу на сервер Wazuh через системний журнал або інтеграцію стороннього API.

Wazuh збирає журнали з різноманітних джерел, забезпечуючи комплексний моніторинг різних аспектів вашого IT-середовища. Ми можемо переглянути нашу документацію щодо збору даних журналу , щоб краще зрозуміти, як Wazuh збирає та аналізує журнали від контрольованих кінцевих точок. Деякі з поширених джерел журналів, які підтримує Wazuh, включають [14]:

*журнали операційної системи*: Wazuh збирає журнали з кількох операційних систем, зокрема Linux, Windows і macOS. Wazuh може збирати syslog, auditd, журнали додатків тощо з кінцевих точок Linux. Wazuh збирає журнали на кінцевих точках Windows за допомогою каналу подій Windows і формату журналу подій Windows. За замовчуванням агент Wazuh відстежує систему, програму та канали подій безпеки Windows на кінцевих точках Windows. Агент Wazuh забезпечує гнучкість налаштування та моніторингу інших каналів подій Windows. Wazuh використовує уніфіковану систему ведення журналів (ULS) для збору журналів на кінцевих точках macOS. macOS ULS централізує керування та зберігання журналів на всіх рівнях системи.

На рисунку 2.6 показано подію, зібрану з каналу подій *Microsoft-Windows-Sysmon/Operational* на кінцевій точці Windows.

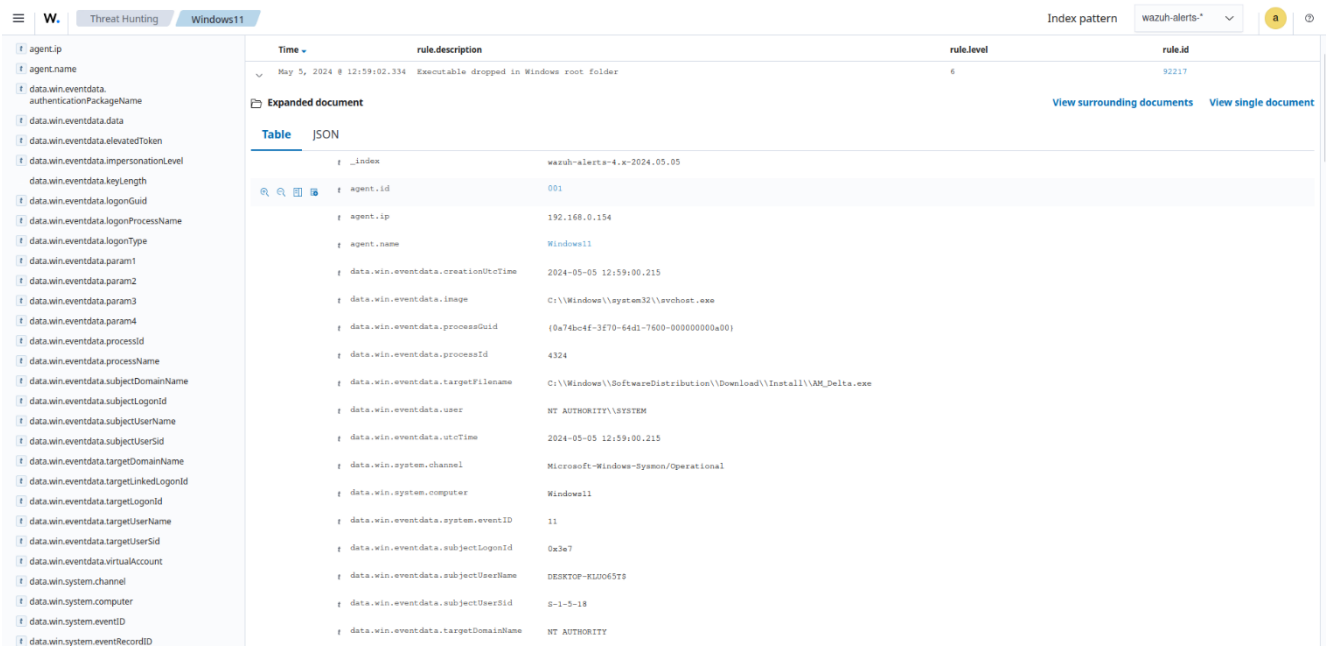


Рис. 2.6. Дані щодо події безпеки в середовищі Wazuh [14]

*події системного журналу:* Wazuh збирає журнали з пристроїв із підтримкою системного журналу, охоплюючи широкий спектр джерел, включаючи системи Linux/Unix і мережеві пристрої, які не підтримують встановлення агента. На рисунку 2.7 показано сповіщення, яке запускається, коли на кінцевій точці Linux створюється новий користувач, а журнал пересилається на сервер Wazuh через *rsyslog*.

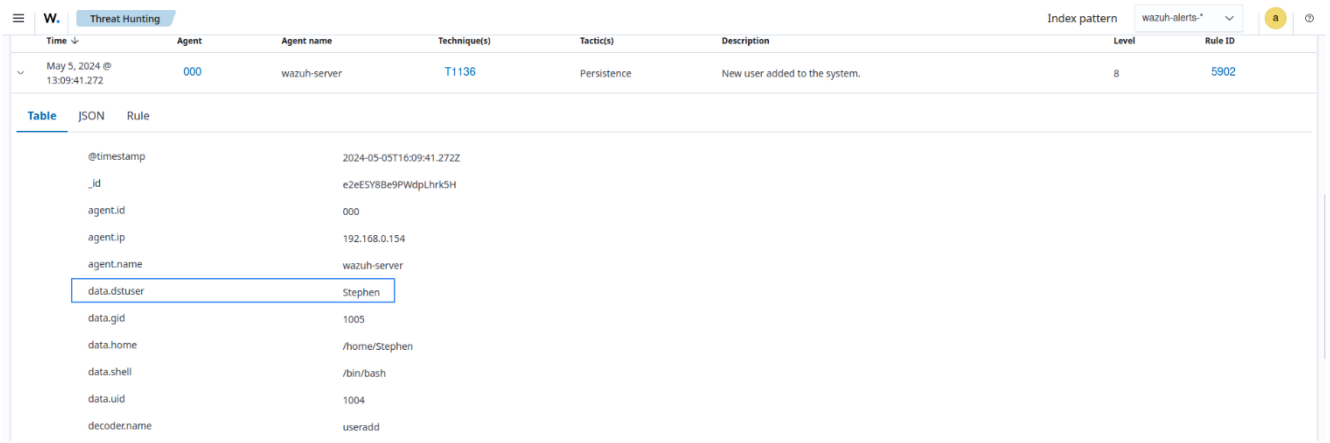


Рис. 2.7. Сповіщення про подію системного журналу

*моніторинг без агента*: Модуль моніторингу без агента Wazuh відстежує кінцеві точки, які не підтримують установку агента. Для цього потрібне з'єднання SSH між кінцевою точкою та сервером Wazuh. Модуль безагентного моніторингу Wazuh відстежує файли, каталоги або конфігурації та виконує команди на кінцевій точці. На рисунку 2.8 показано сповіщення від пристрою без агента на інформаційній панелі Wazuh.

*журнали хмарних постачальників*: Wazuh інтегрується з хмарними постачальниками, такими як AWS, Azure , Google Cloud і Office 365, щоб збирати журнали з хмарних служб, таких як екземпляри EC2, сегменти S3, віртуальні машини Azure тощо. На рисунку 2.9 показано розділ «*Хмарна Безпека*» на інформаційній панелі Wazuh.

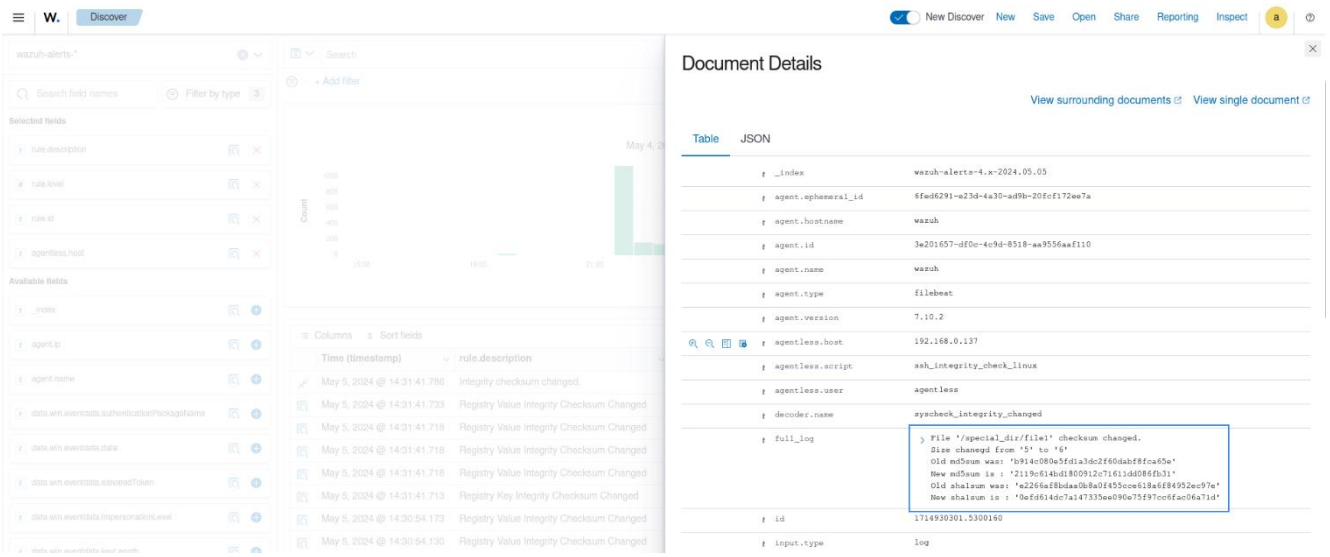


Рис. 2.8. Сповіщення від пристрою без агента на інформаційній панелі Wazuh

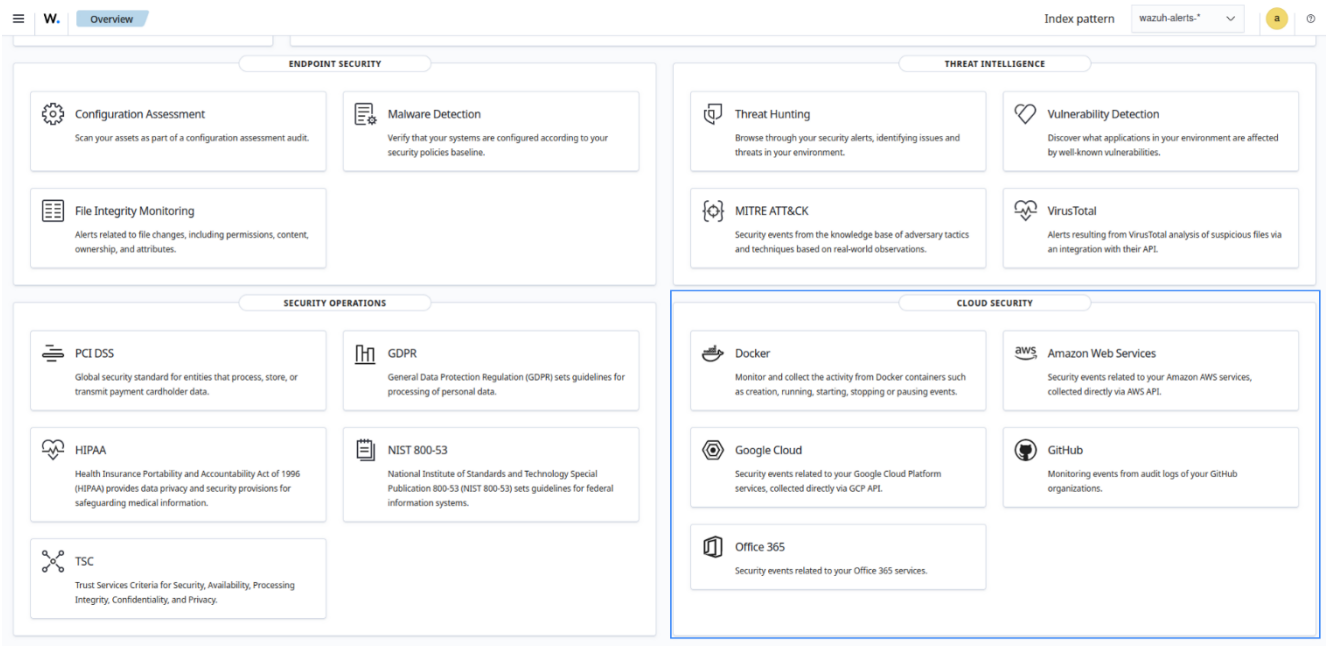


Рис. 2.9. Панель «Хмарна Безпека» на інформаційній панелі Wazuh

*спеціальні журнали*: ми можемо налаштувати Wazuh для збору та аналізу журналів із кількох програм і сторонніх інструментів безпеки, таких як VirusTotal, Windows Defender і ClamAV. На рисунку 2.10 показано сповіщення про журнал із VirusTotal, оброблений сервером Wazuh.


Рис. 2.10. Сповіщення про журнал із VirusTotal

*Правила та декодери* Wazuh є основними компонентами аналізу даних журналу, виявлення загроз і реагування на них. Wazuh надає потужну платформу для аналізу даних журналу, що дозволяє організаціям підвищити рівень безпеки шляхом оперативного виявлення потенційних загроз безпеці та реагування на них.

Декодери Wazuh відповідають за аналіз та нормалізацію даних журналу, зібраних із різних джерел. Декодери необхідні для перетворення необроблених даних журналу в кількох форматах в єдиний і структурований формат, який Wazuh може ефективно обробляти. У Wazuh є готові декодери для поширених форматів журналів, таких як *syslog*, канал подій Windows, macOS ULS тощо. Крім того, Wazuh дозволяє визначати спеціальні декодери для аналізу журналів із певних програм або пристроїв з унікальними форматами журналів. Використовуючи декодери, Wazuh може ефективно інтерпретувати дані журналу та отримувати відповідну інформацію, таку як мітки часу, рівні журналу, вихідні IP-адреси, імена користувачів тощо.

*Набір правил Wazuh* виявляє події безпеки та аномалії в даних журналу. Ці правила написані в спеціальному форматі, і вони запускають сповіщення, коли виконуються певні умови. Правила визначаються на основі певних критеріїв, таких як поля журналу, значення або шаблони, щоб відповідати певним записам журналу, які можуть вказувати на загрози безпеці. Wazuh надає широкий спектр готових правил, які охоплюють типові випадки використання безпеки. Крім того, адміністратори можуть створювати власні правила, адаптовані до їхнього

середовища та вимог безпеки.

Вразливості – це недоліки безпеки в комп'ютерних системах, якими зловмисники можуть скористатися для отримання несанкціонованого доступу до цих систем. Після експлуатації шкідливе програмне забезпечення та загрози можуть виконувати дистанційне виконання коду, викрадати дані та виконувати інші шкідливі дії. Тому організації повинні мати стратегії або рішення безпеки, які оперативно виявляють уразливості в їхній мережі, перш ніж зловмисники використають їх. Швидке виявлення та усунення вразливостей у мережі допомагає зміцнити її загальну безпеку.

*Модуль виявлення вразливостей Wazuh* допомагає користувачам виявляти вразливості в операційній системі та програмах, встановлених на контрольованих кінцевих точках. Модуль працює за допомогою одного з наступних джерел вразливості [14]:

сховище вразливостей Wazuh на платформі Cyber Threat Intelligence (CTI);

офлайн-локальне сховище вразливостей.

Фахівці Wazuh збирають дані про вразливості із зовнішніх джерел вразливостей, індексованих Canonical, Debian, Red Hat, Arch Linux, Amazon Linux Advisories Security (ALAS), Microsoft, CISA та Національною базою даних вразливостей (NVD), щоб надати інформацію про вразливості. Дані CISA мають пріоритет над NVD, щоб забезпечити точніше та надійніше виявлення активно використовуваних вразливостей. Фахівці Wazuh постійно оновлюють цю інформацію, забезпечуючи перевірку рішення на наявність останніх CVE.

# 3 РЕКОМЕНДАЦІЇ ЩОДО ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ІНЦИДЕНТИ БЕЗПЕКИ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА БАЗІ WAZUH

## 3.1. Порядок виявлення та реагування на інциденти безпеки в інформаційній системі організації із застосуванням рішення Wazuh

*Розглянемо порядок збирання даних журналу в інформаційній системі організації та реагування на них із застосуванням рішення Wazuh.*

Збір даних журналу передбачає збирання та консолідацію журналів із різних джерел журналів у мережі. Збір даних журналу допомагає групам безпеки відповідати нормативним вимогам, виявляти та усувати загрози, а також визначати помилки програм та інші проблеми безпеки.

Wazuh збирає, аналізує та зберігає журнали з кінцевих точок, мережевих пристроїв і програм. Агент Wazuh, який працює на контрольованій кінцевій точці, збирає та пересилає журнали системи та програм на сервер Wazuh для аналізу. Крім того, ми можемо надсилати повідомлення журналу на сервер Wazuh через системний журнал або сторонні інтеграції API [15].

Wazuh використовує модуль Logcollector для збору журналів із контрольованих кінцевих точок, програм і мережевих пристроїв. Потім сервер Wazuh аналізує зібрані журнали в режимі реального часу за допомогою декодерів і правил. Wazuh витягує відповідну інформацію з журналів і відображає їх у відповідних полях за допомогою декодерів. Модуль Analysisd на сервері Wazuh оцінює декодовані журнали відповідно до правил і записує всі сповіщення *в /var/ossec/logs/alerts/alerts.logфайли/var/ossec/logs/alerts/alerts.json*.

Окрім журналів сповіщень, Wazuh зберігає всі зібрані журнали у спеціальних архівних файлах журналу, зокрема */var/ossec/logs/archives/archives.logma /var/ossec/logs/archives/archives.json*. Ці файли архівних журналів повністю фіксують усі журнали, включно з тими, які не

викликають жодних сповіщень. Ця функція забезпечує повний запис усіх дій системи для подальшого використання та аналізу.

Сервер Wazuh також отримує повідомлення системного журналу від пристроїв, які не підтримують установку агентів Wazuh, забезпечуючи бездоганну інтеграцію та покриття в усьому мережевому середовищі.

За замовчуванням сервер Wazuh зберігає журнали і не видаляє їх автоматично. Однак ми можемо вибрати, коли видаляти ці журнали вручну чи автоматично відповідно до нормативних вимог.

На рисунку 3.1. показано процес збору та аналізу даних журналу в Wazuh.

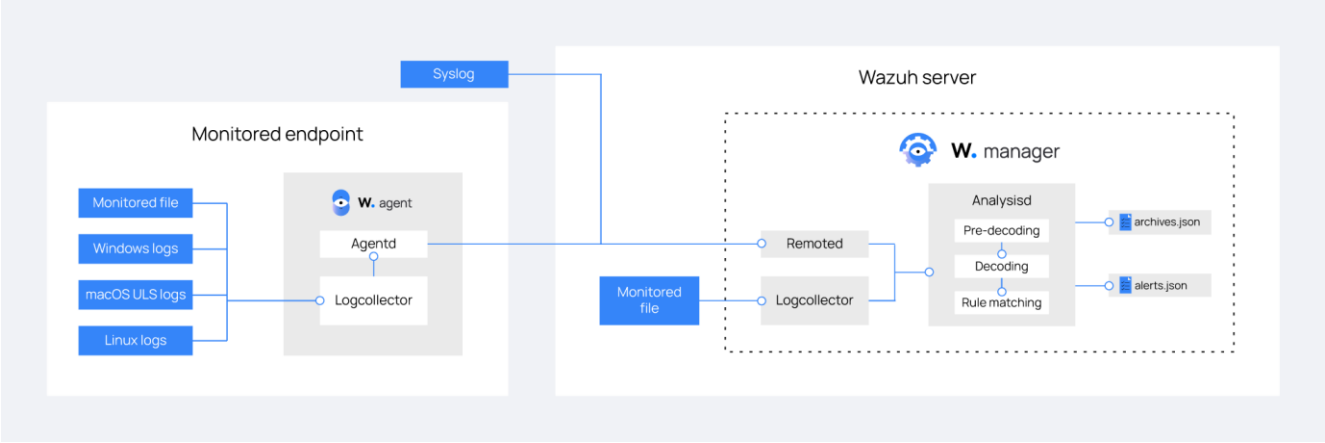


Рис. 3.1. Процес збору та аналізу даних журналу в Wazuh [15]

*Розглянемо порядок виявлення вразливостей в інформаційній системі організації та реагування на них із застосуванням рішення Wazuh.*

Щоб виявити вразливості, агенти Wazuh збирають список встановлених додатків із контрольованих кінцевих точок і періодично надсилають його на сервер Wazuh. Цей список зберігається в локальних базах даних SQLite на сервері Wazuh. На сервері Wazuh модуль виявлення вразливостей співвідносить дані інвентаризації програмного забезпечення з документами вмісту вразливостей, щоб виявити вразливе програмне забезпечення на контрольованій кінцевій точці. Ці документи є записами про загальні вразливості та викриття (CVE), які доступні на платформі аналізу кіберзагроз (CTI) Wazuh [15].

На платформі CTI Wazuh збираються дані про вразливості з різних джерел,

як-от постачальників операційних систем і баз даних вразливостей, об’єднуючи їх у єдине надійне сховище. Процес передбачає стандартизацію різноманітних форматів у загальну структуру за допомогою формату CVE JSON 5.

Щоб оновити інформацію про вразливість на сервері Wazuh, модуль виявлення вразливості запитує CTI API або автономне локальне сховище. Він отримує нові документи та порівнює їх із існуючими, щоб виявити будь-які відмінності [15].

Далі модуль сканує інвентар програмного забезпечення кінцевих точок, використовуючи останню доступну інформацію про вразливості. Процес виявлення шукає вразливі пакети в базах даних інвентаризації. Ці запаси унікальні для кожного агента.

Пакет позначається як вразливий, якщо його версія потрапляє в уражений діапазон CVE. Модуль генерує сповіщення для відображення результатів, а результати зберігаються в загальному списку вразливостей. Цей інвентар можна фільтрувати за агентом і містить лише невирішені вразливості. Користувачі можуть запитувати інвентаризацію , щоб переглянути сповіщення та подробиці про вразливості [15].

Для систем Microsoft Windows і певних продуктів Microsoft модуль виявлення вразливостей містить опцію виправлень у налаштуваннях Syscollector агента Wazuh. Увімкнення цієї опції дозволяє модулю виявляти пакети, які користувачі виправили. Коли виявляється виправлення, модуль використовує інформацію, надану Microsoft, щоб перевірити, чи виправлення усуває відповідні CVE. Якщо це так, модуль видаляє ці вразливості зі списку.

*Генерація сповіщень*. Модуль виявлення вразливостей генерує сповіщення, коли він виявляє нові вразливості або коли вразливість виправлено через оновлення пакета, видалення чи оновлення системи. Однак для підвищення надійності платформи Wazuh згадані вище умови є необхідними, але недостатніми для генерації сповіщень. Сповіщення вказує на те, що щойно щось сталося, але це не завжди так [15]:

*сповіщення ОС*: вони генеруються не під час початкового сканування, а під

час наступних сканувань. Якщо агент синхронізується з менеджером уперше, він не реєструє, що версія ОС змінилася або нещодавно було застосовано виправлення;

*сповіщення про пакети*: сповіщення спрацьовують, коли інсталяція або видалення пакета призводить до зміни списку вразливостей. Однак це відбувається лише тоді, коли агент виявляє подію під час звичайного сканування Syscollector. Зміни, внесені під час зупинки агента або примусового повідомлення про нього через перезапуск, не створюють сповіщень.

Інші фактори, які слід враховувати щодо створення сповіщень, включають [15]:

*середовище кластера*: якщо агент переходить на інший хост, його інвентар синхронізується з новим хостом, але під час цієї початкової синхронізації сповіщення не генеруються;

*оновлення вмісту*: коли вміст змінюється, усі агенти скануються повторно, щоб переконатися, що результати актуальні, але сповіщення не генеруються під час цієї початкової синхронізації.

Схема цього робочого процесу показано на рисунку 3.2.

Враховуючи всі можливі ситуації, найнадійнішим способом визначення поточних вразливостей у середовищі є запит до списку вразливостей [15].

*Розглянемо порядок реагування на інциденти в інформаційній системі організації із застосуванням рішення Wazuh.*

Інцидент безпеки стосується будь-якої несприятливої події чи діяльності, яка ставить під загрозу конфіденційності, цілісності або доступності цифрових активів, мереж, даних або ресурсів. Такі інциденти включають несанкціонований доступ, витік даних, зараження шкідливим програмним забезпеченням, атаки на відмову в обслуговуванні та будь-які інші дії, які ставлять під загрозу безпеку інформаційно-технологічного середовища організації [15].

Мета реагування на інциденти полягає в тому, щоб ефективно впоратися з інцидентом безпеки та якомога швидше відновити нормальну роботу бізнесу. Оскільки цифрові активи організацій постійно зростають, управління

інцидентами вручну стає дедалі складнішим, отже, потреба в автоматизації.

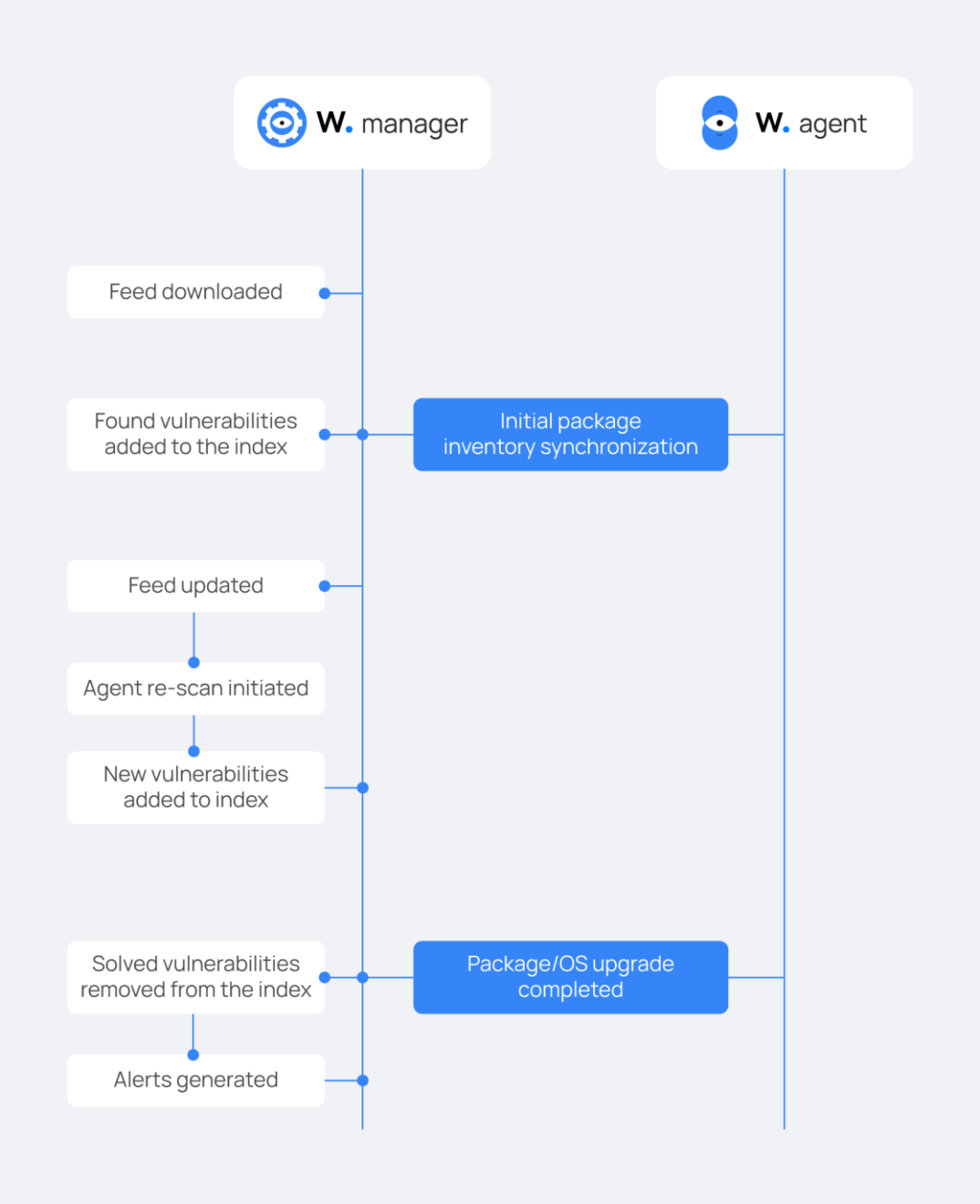


Рис. 3.2. Схема робочого процесу виявлення вразливостей в інформаційній системі організації та реагування на них із застосуванням рішення Wazuh

Автоматизоване реагування на інциденти передбачає автоматичні дії, які виконуються під час реагування на інциденти безпеки. Ці дії можуть включати ізоляцію скомпрометованих кінцевих точок, блокування шкідливих IP-адрес, розміщення заражених пристроїв на карантині або вимкнення скомпрометованих облікових записів користувачів. Автоматизуючи реагування на інциденти, групи

кібербезпеки скорочують час реагування на виявлені загрози, запобігають або мінімізують вплив інцидентів і ефективно обробляють велику кількість подій безпеки.

*Модуль Wazuh Active Response* дозволяє користувачам запускати автоматизовані дії при виявленні інцидентів на кінцевих точках. Це покращує процеси реагування на інциденти в організації, дозволяючи командам безпеки вживати негайних автоматизованих дій для протидії виявленим загрозам.

Ми також можемо налаштувати дії як без стану, так і без стану. Активні відповіді без стану є одноразовими діями, тоді як відповіді зі збереженням стану повертають свої дії через деякий час.

*Активні відповідні дії за замовчуванням*

Готові сценарії доступні в кожній операційній системі, яка запускає агенти Wazuh. Деякі зі стандартних сценаріїв активної відповіді показано в таблиці 3.1.

Таблиця 3.1

Деякі сценарії активної відповіді

| Назва сценарію | опис |
|---|---|
| disable-account | Відключає обліковий запис користувача |
| firewall-drop | Додає IP-адресу до списку заборонених iptables. |
| firewall-drop | Додає IP-адресу до спадного списку брандмауера. |
| restart.sh | Перезапускає агент або сервер Wazuh. |
| netsh.exe | Блокує IP-адресу за допомогою netsh. |

*Настроювані дії активної відповіді*

Однією з переваг модуля Wazuh Active Response є його адаптивність. Wazuh дозволяє командам безпеки створювати власні активні дії відповіді на будь-якій мові програмування, пристосовуючи їх до своїх конкретних потреб. Це гарантує,

що при виявленні загрози відповідь може бути налаштована відповідно до вимог організації.

*Автоматизація реагування на інциденти за допомогою Wazuh*

Щоб використовувати модуль Wazuh Active Response, нам потрібно налаштувати дію, яка буде виконуватися, коли на контрольованій кінцевій точці відбувається певна подія. Наприклад, ми можемо налаштувати модуль Wazuh Active Response на видалення шкідливого виконуваного файлу із зараженої кінцевої точки. У наведених нижче прикладах ми показуємо, як модуль Wazuh Active Response обробляє різні інциденти.

*Видалення шкідливих програм*

Ми можемо використовувати модуль Wazuh Active Response у поєднанні з модулем моніторингу цілісності файлів та інтеграцією VirusTotal для виявлення та видалення шкідливих файлів із кінцевої точки.

На рисунку 3.3 показані такі дії:

1. Ідентифікатор правила 554 запускається, коли файл додається до каталогу Downloads, який відстежується за допомогою модуля моніторингу цілісності файлів Wazuh.

2. Ідентифікатор правила 87105 спрацьовує, коли Wazuh витягує хеш файлу, запитує дані про хеш файлу з бази даних VirusTotal через свій API та отримує відповідь на шкідливий файл.

3. Ідентифікатор правила 55 запускається, коли файл видаляється з каталогу Downloads, який відстежується за допомогою модуля моніторингу цілісності файлів Wazuh.

4. Ідентифікатор правила 11000 запускається, коли модуль Wazuh Active Response видаляє шкідливий файл із кінцевої точки.

У цьому сценарії модуль Wazuh Active Response автоматично видаляє шкідливий файл, скорочуючи час між виявленням загрози та пом’якшенням.

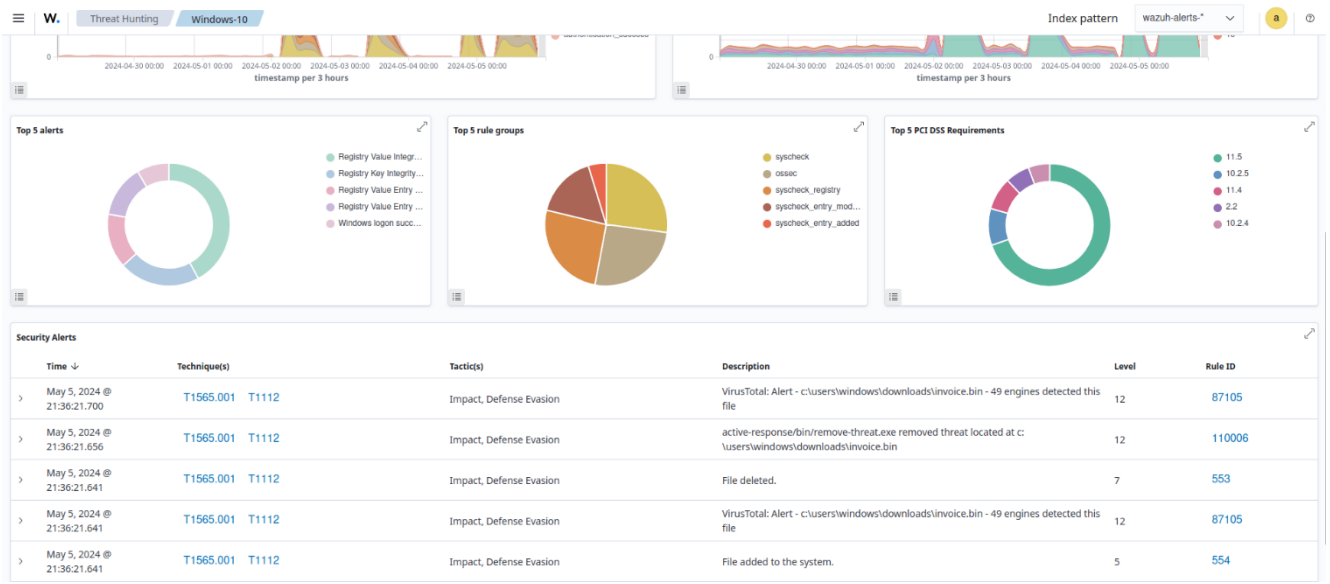


Рис. 3.3. Дії з видалення шкідливих програм

*Відповідь на DoS-атаки*

Основна мета DoS-атаки полягає в тому, щоб зробити ціль недоступною для законних користувачів, викликаючи відмову в обслуговуванні. На рисунку 3.4 показано, як модуль Wazuh Active Response блокує шкідливі IP-адреси, які виконують DoS проти веб-сервера на кінцевій точці Ubuntu.

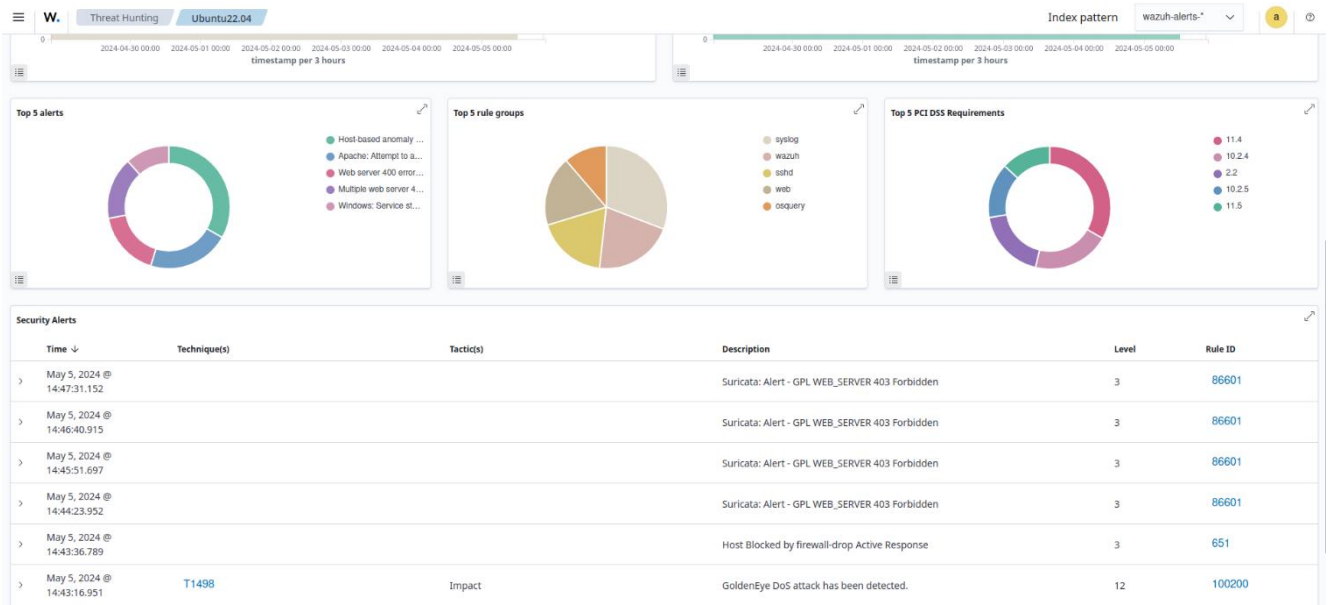


Рис. 3.4. Відповідь на DoS-атаки

У цьому випадку модуль Wazuh Active Response автоматично блокує

шкідливі хости від виклику DoS-атаки на веб-сервері. Таким чином забезпечується доступність веб-сервера для авторизованих користувачів.

*Вимкнення облікового запису користувача після атаки грубої сили*

Блокування облікового запису – це захід безпеки, який використовується для захисту від атак грубої сили шляхом обмеження кількості спроб входу, які користувач може зробити протягом визначеного часу. Ми використовуємо модуль Wazuh Active Response, щоб вимкнути обліковий запис користувача, пароль якого вгадується зловмисником.

На рисунку 3.5 показано, як модуль Wazuh Active Response вимикає обліковий запис на кінцевій точці Linux і знову вмикає його через 5 хвилин.

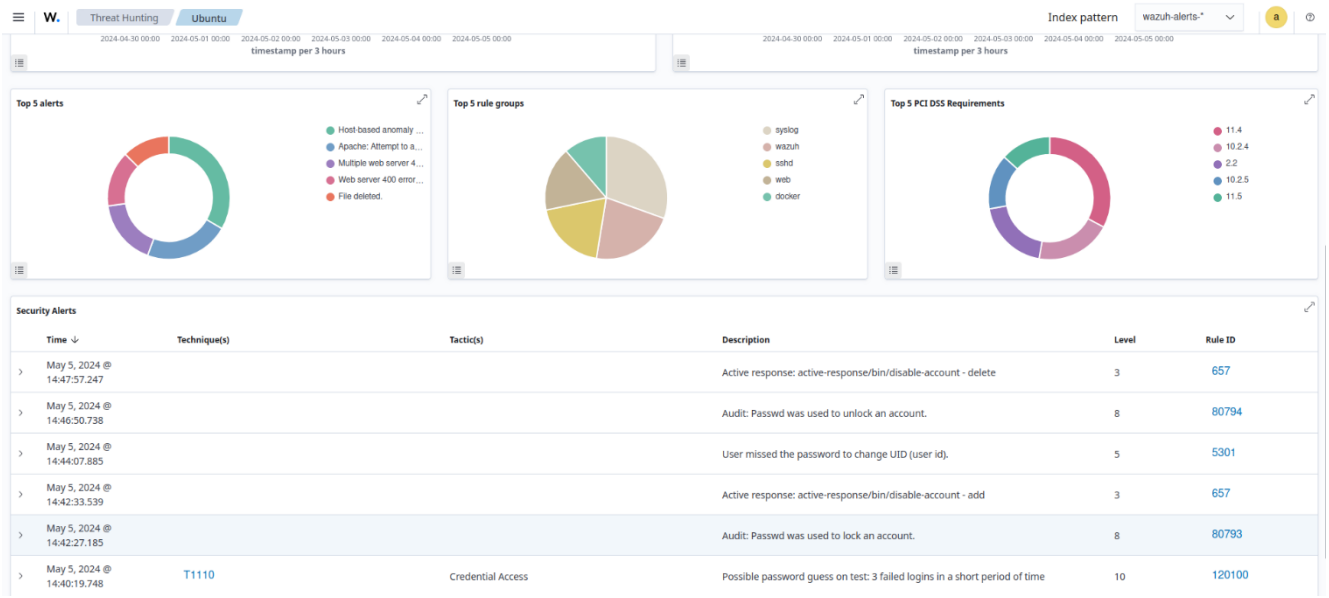


Рис. 3.5. Вимкнення облікового запису користувача після атаки грубої сили

У цьому сценарії, коли зловмисник неодноразово намагається вгадати пароль користувача та не вдається, обліковий запис стає тимчасово недоступним. Це заважає зловмисникам, які покладаються на методи грубої сили, щоб вгадати паролі облікових записів користувачів.

Використовуючи модуль Wazuh Active Response, служби безпеки можуть автоматизувати реагування на різні інциденти. Таким чином забезпечується ефективне реагування на інциденти та більш стійка кібербезпека.

## 3.2. Рекомендації щодо виявлення та реагування на інциденти безпеки в інформаційній системі організації

Процеси виявлення та реагування на інциденти безпеки в організаціях часто стикаються з низкою проблем, які можуть суттєво впливати на їхню ефективність. Однією з поширених проблем є великий обсяг попереджень та хибно-позитивних спрацьовувань від систем безпеки. Організації можуть отримувати тисячі або навіть мільйони сповіщень на день, багато з яких виявляються нешкідливими. Це призводить до так званої «втоми від попереджень», коли аналітики безпеки починають ігнорувати сповіщення через їхню велику кількість, що може призвести до пропуску реальних інцидентів.

Іншою серйозною проблемою є складність виявлення складних та прихованих загроз, таких як цілеспрямовані атаки (APT) та інсайдерські загрози. Ці загрози часто використовують новітні методи та техніки, що дозволяє їм залишатися непоміченими протягом тривалого часу. Інсайдерські загрози особливо важко виявити, оскільки зловмисники вже мають легітимний доступ до інформаційних систем організації.

Брак кваліфікованих фахівців та ресурсів є ще однією значною перешкодою для ефективного виявлення та реагування на інциденти безпеки. Кібербезпека є швидкозмінною сферою, що вимагає постійного навчання та підвищення кваліфікації, і багато організацій відчувають нестачу фахівців з необхідними знаннями та досвідом. Обмежені бюджети також можуть обмежувати можливості організацій у впровадженні сучасних технологій та залученні зовнішніх експертів.

Недостатня видимість та фрагментація даних в інформаційній інфраструктурі організації також ускладнюють виявлення та аналіз інцидентів. Дані про безпеку можуть бути розкидані по різних системах та пристроях, що ускладнює їхню централізацію та кореляцію для отримання цілісної картини інциденту.

Нарешті, тривалий час виявлення та локалізації інцидентів є серйозною проблемою. Згідно з дослідженнями, середній час виявлення та локалізації

інциденту може становити від кількох тижнів до кількох місяців. За цей час зловмисники можуть завдати значної шкоди організації, викрасти конфіденційні дані або порушити роботу критично важливих систем.

Для підвищення ефективності виявлення та реагування на інциденти безпеки організації повинні дотримуватися низки найкращих практик. Однією з найважливіших є розробка та регулярне оновлення *Планів реагування на інциденти*. Планів реагування на інциденти повинні містити чіткі інструкції щодо дій на кожному етапі процесу реагування, а також визначати ролі та обов'язки членів команди реагування. Регулярне тестування та оновлення планів реагування є критично важливим для забезпечення їхньої актуальності та ефективності.

Формування та навчання *команди реагування на інциденти* є ще однією ключовою найкращою практикою. Команда повинна складатися з кваліфікованих фахівців з різних областей кібербезпеки, які мають чітке розуміння своїх ролей та обов'язків. Регулярні навчання та тренування, включаючи симуляції реальних інцидентів, допомагають команді підтримувати високий рівень готовності та вдосконалювати свої навички.

Використання автоматизованих інструментів виявлення та реагування, таких як SIEM, SOAR (Security Orchestration, Automation and Response), EDR (Endpoint Detection and Response) та XDR (Extended Detection and Response), є важливим для підвищення швидкості та ефективності обробки інцидентів. Ці інструменти допомагають автоматизувати збір та аналіз даних безпеки, виявляти аномалії та загрози, а також оркеструвати процеси реагування.

Рекомендується також проводити регулярні навчання та тренування не лише для команди реагування, але й для всього персоналу організації. Підвищення обізнаності користувачів щодо кіберзагроз та правил безпечної поведінки може значно зменшити кількість успішних атак, особливо фішингових.

Нарешті, обмін інформацією про загрози та інциденти з іншими організаціями та галузевими спільнотами може допомогти отримати уявлення про актуальні загрози, тактики зловмисників та найкращі практики реагування. Участь у таких обмінах інформацією сприяє підвищенню загального рівня

кібербезпеки в галузі.

Звітність про інциденти безпеки є важливим аспектом процесу реагування. Детальне документування всіх етапів процесу реагування, від виявлення до відновлення, є критично важливим для відстеження ходу дій, аналізу причин та наслідків інциденту, а також для виявлення вивчених уроків.

Звіт про інцидент безпеки повинен мати чітку структуру та включати такі ключові елементи, як опис інциденту, хронологія подій, оцінка впливу на бізнес, перелік вжитих заходів з реагування, а також висновки та рекомендації щодо запобігання подібних інцидентів у майбутньому. Наявність стандартизованої структури звіту полегшує його аналіз та використання для покращення процесів інформаційної безпеки.

Організації також повинні враховувати вимоги до звітності про інциденти безпеки, встановлені різними регуляторними нормами, такими як GDPR, PCI DSS та інші. Дотримання цих вимог є важливим для уникнення штрафів та юридичних наслідків, а також для підтримання довіри з боку клієнтів та партнерів.

Отже, проведене дослідження підтверджує, що проблема виявлення та реагування на інциденти безпеки є однією з найважливіших у сфері кібербезпеки. Зростання кількості та складності кіберзагроз вимагає від організацій не лише постійного вдосконалення засобів захисту, але й створення ефективних механізмів для своєчасного виявлення інцидентів та адекватного реагування на них.

Результати дослідження підкреслюють важливість комплексного підходу до виявлення та реагування на інциденти безпеки. Організації повинні не лише впроваджувати сучасні технології моніторингу та аналізу, але й розробляти чіткі плани реагування, формувати кваліфіковані команди, проводити регулярні навчання та обмінюватися інформацією про загрози. Детальне документування всіх етапів процесу реагування та врахування вимог регуляторних органів також є важливими складовими ефективної системи управління інцидентами безпеки.

# ВИСНОВКИ

В роботі досліджено проблему виявлення та реагування на інциденти безпеки в інформаційній системі організації. Встановлено, що виявлення та реагування на інциденти безпеки є невід'ємною частиною забезпечення інформаційної безпеки організації, що допомагає захистити її активи, підтримувати безперервність бізнесу, дотримуватися законодавчих вимог та зберігати довіру зацікавлених сторін. Ефективне реагування на інциденти дозволяє швидко локалізувати загрозу, відновити працездатність систем та мінімізувати час простою.

Визначено існуючі підходи до виявлення та реагування на інциденти безпеки в інформаційній системі. Процес виявлення інцидентів безпеки є критично важливим етапом у загальному циклі реагування на інциденти. Ефективне виявлення дозволяє організації своєчасно ідентифікувати потенційні загрози та вжити необхідних заходів для мінімізації їхнього впливу. Різні фреймворки, такі як NIST та SANS, пропонують структуровані підходи до процесу виявлення та реагування на інциденти безпеки.

Проаналізовано існуючі рішення з виявлення та реагування на інциденти безпеки в інформаційній системі. SIEM-системи відіграють вирішальну роль у сучасній кібербезпеці для організацій будь-якого розміру. SIEM-системи агрегують та аналізують дані з різноманітних інструментів безпеки та IT. Вони перетворюють події та журнали на корисну інформацію для виявлення загроз, реагування на інциденти та забезпечення відповідності нормативним вимогам. Зважаючи на зростаючу складність кібератак та посилення вимог регуляторних органів, впровадження SIEM-рішень стає необхідністю.

Рішення Wazuh – це централізована платформа для агрегування й аналізу телеметрії в режимі реального часу для виявлення загроз і відповідності. Wazuh збирає дані про події з різних джерел, таких як кінцеві точки, мережеві пристрої, хмарні робочі навантаження та програми для ширшого охоплення безпеки.

В роботі розглянуто методи та засоби виявлення та реагування на інциденти безпеки в інформаційній системі організації із застосуванням рішення Wazuh. Рішення Wazuh базується на агенті Wazuh, який розгортається на контрольованих кінцевих точках, і на трьох центральних компонентах: сервері Wazuh, індексаторі Wazuh і інформаційній панелі Wazuh.

В роботі розглянуто порядок виявлення та реагування на інциденти безпеки в інформаційній системі організації із застосуванням рішення Wazuh. Розглянуто порядок збирання даних журналу в інформаційній системі організації та реагування на них із застосуванням рішення Wazuh. Розглянуто порядок виявлення вразливостей в інформаційній системі організації та реагування на них із застосуванням рішення Wazuh. Розглянуто порядок реагування на інциденти в інформаційній системі організації із застосуванням рішення Wazuh.

Розроблено рекомендації фахівцям з кібербезпеки щодо виявлення та реагування на інциденти безпеки в інформаційній системі організації.

Таким чином, правильне застосування методів та засобів виявлення та реагування на інциденти безпеки має забезпечити ефективний захист інформаційних ресурсів організації.

# ПЕРЕЛІК ПОСИЛАНЬ


1. ENISA Threat Landscape 2024. Publication date: September 19, 2024. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

2. 2024 Data Breach Investigations Report. URL: https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf

3. Lee, Y.-J. A Study on a Scenario-Based Security Incident Prediction System for Cybersecurity. Appl. Sci. 2024, 14, 11836. https://doi.org/10.3390/app142411836

4. What is Incident Response? IBM. URL: https://www.ibm.com/think/topics/incident-response

5. security incident – Glossary, CSRC. URL: https://csrc.nist.gov/glossary/term/security_incident

6. What is Incident Response? Process, Frameworks, and Tools. BlueVoyant. URL: https://www.bluevoyant.com/knowledge-center/what-is-incident-response-process-frameworks-and-tools

7. ISO 27001. Information security event vs. incident vs. non-compliance – Advisera. URL: https://advisera.com/27001academy/blog/2018/12/03/iso-27001-information-security-event-vs-incident-vs-non-compliance/

8. Alex Nelson, Sanjay Rekhi, Murugiah Souppaya, Karen Scarfone. Incident Response Recommendations and Considerations for Cybersecurity Risk Management A CSF 2.0 Community Profile. NIST SP 800-61r3. This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.800-61r3

9. SANS Incident Response: 6-Step Process & Critical Best Practices. Exabeam. URL: https://www.exabeam.com/explainers/incident-response/sans-incident-response-6-step-process-critical-best-practices/

10. What is IDS and IPS? Juniper Networks US. URL: https://www.juniper.net/us/en/research-topics/what-is-ids-ips.html

11. Orion Cassetto. 7 Open Source SIEMs: Functions and Limitations. Macnica.

URL: https://www.macnica.co.jp/en/business/security/manufacturers/exabeam/feature_15.html

12. Avinash Kumar. ELK stack - Log Analysis and Monitoring. 28 липня 2023 р. URL: https://www.linkedin.com/pulse/elk-stack-log-analysis-monitoring-avinash-kumar/

13. Austin Mitchell. 8 Open Source SIEM Tools You Should Know. June 11, 2024. URL: https://www.lumificyber.com/blog/8-open-source-siem-tools-you-should-know/

14. Wazuh. Getting started with Wazuh. Version 4.11. URL: https://documentation.wazuh.com/current/getting-started/index.html

15. Wazuh. User manual. Version 4.11. URL: https://documentation.wazuh.com/current/user-manual/index.html

16. Wazuh. A comprehensive SIEM solution. URL: https://wazuh.com/platform/siem/

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

**(Презентація)**