МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

Фляшко Назарій Русланович

Алгоритми виявлення шкідливого програмного забезпечення за допомогою Wazuh / Malware Detection Algorithms Using Wazuh

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
Н.Р. Фляшко

Науковий керівник
к.т.н., доцент Н.Г.Яцків

Кваліфікаційну роботу
допущенодо захисту:

«____» _____ ___2024 р.

Завідувач кафедри
_____**В.В.Яцків**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій

Кафедра кібербезпеки

Освітній ступінь «магістр»

спеціальність: 125 –Кібербезпека

освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
«____» _____ 20___ року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Фляшко Назарій Русланович

(прізвище, ім'я по-батькові)

1. Тема кваліфікаційної роботи:
Алгоритми виявлення шкідливого програмного забезпечення за допомогою Wazuh / Malware Detection Algoritms Using Wazuh
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 12 грудня 2023 року № 753

2. Строк подання студентом закінченої кваліфікаційної роботи 12 грудня 2024р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
   - дослідити роботу платформи Wazuh;
   - зробити моніторинг системи;
   - ефективно виявити та зреагувати на загрози шкідливого забезпечення;
   - зменшити ризики, що можуть виникнути внаслідок не вчасного реагування на інциденти;
   - розробити алгоритмизахистуWazuh в центрі досліджуваної системи.

5. Перелік графічного матеріалу у роботі:
   - архітектура машини Wazuh;
   - структура агентаWazuh;
   - інтерфейс платформи Wazuh;

- додавання агента в операційну систему віртуальної машини;
- результати виявлених загроз агентом.

6. Консультанти розділів випускної кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз систем виявлення та запобігання вторгнень	12.2023р. – 03.2024р.	
2	Структура досліджуваного середовища Wazuh	03.2024р. – 06.2024р.	
3	Дослідження та реалізація виявлення загроз машиною Wazuh	06.2024р. – 11.2024р.	

Студент _____ (підпис) Фляшко Н.Р.

Керівник роботи _____ (підпис) к.т.н., доцент Н.Г. Яцків

АНОТАЦІЯ

Магістерська робота на тему «Алгоритми виявлення шкідливого програмного забезпечення за допомогою Wazuh» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека та захист інформації» освітньо-професійної програми «Кібербезпека» написана обсягом 85 сторінок та містить 36 зображень, 3 таблиці та 32 джерела за переліком посилань.

Мета кваліфікаційної роботи полягає в тестуванні систем виявлення та запобігання вторгнень (IDS/IPS) в корпоративної мережі, що застосовується для посилення захисту від кіберзагроз шляхом активного моніторингу мережевого трафіку, виявляючи потенційні загрози та запобігаючи несанкціонованому доступу або зловмисним діям.

В якості технічного обладнання що спеціалізується на вирішені питань безпеки було вибрано систему Wazuh, яка відповідає за збір даних за допомогою агента, обробляє та аналізує дані на серверній частині, а також надсилає результати роботи активної відповіді на атаки спричинені шкідливим програмним забезпечення на відповідні інформаційні джерела такі як електронна пошта або інші.

Таким чином було реалізовано клієнт-серверну мережу оповіщень, на боці сервера виступатиме локальне Wazuh середовище, а за клієнтську складову відповідатиме поштовий відділ Gmail. Завдяки тригерам було налаштовано робочу область Wazuh в рамках якої він працюватиме. Внесено правило для в конфігураційний файл агента для того щоб він зосередив увагу на конкретному шкідливому програмному забезпеченні в межах якого його було розроблено.

ABSTRACT

The master's work on the topic “Algorithms for detecting malware using Wazuh” for the degree of Master's in specialty 125 “Cybersecurity and Information Protection” of the educational and professional program “Cybersecurity” is written in 85 pages and contains 36 images, 3 tables and 32 sources in the list of references.

The purpose of the qualification work is to test intrusion detection and prevention systems (IDS/IPS) in a corporate network, which is used to strengthen protection against cyber threats by actively monitoring network traffic, identifying potential threats and preventing unauthorised access or malicious actions.

Wazuh was chosen as the technical equipment specialising in security issues, which is responsible for collecting data using an agent, processing and analysing data on the server side, and sending the results of an active response to attacks caused by malware to the relevant information sources such as e-mail or others.

Thus, a client-server alert network was implemented, with the local Wazuh environment acting as the server side, and the Gmail mail department being responsible for the client component. Thanks to the triggers, the Wazuh workspace was configured within which it will work. A rule was added to the agent's configuration file to make it focus on the specific malware within which it was developed.

ЗМІСТ

ВСТУП

Актуальність виявлення шкідливого програмного забезпечення відіграє головну роль в ієрархічній структурі безпеки цифрової системи та даних. Під словом шкідливе розуміється зловмисне програмне забезпечення, що має на меті завдати шкоду конфіденційній інформації, порушити цілісність обробки, та можливість отримання доступу до інформаційного ресурсу.

Вплив шкідливого програмного забезпечення несе небезпеку для людей, фінансових активів, та й усього суспільства. Його діапазон може варіюватися від фінансових витоків до завдання шкоди репутації організації. Зловмисне програмне забезпечення може викликати крадіжку особистих даних, шахрайство, та порушення захищеності інформації, що викликає збої в системі, уповільнює роботу системи перевантажуючи її запитами та може призводити до поломки приладу. Сукупність таких чинників сприяє продукуванню кібератак, до якого відносяться: DDoS, програми-вимагачі, та фішингові атаки.

Однак люди все частіше довіряють свою інформацію різноманітним комп'ютерним системам, які автоматизують їхнє звичне життя, роблячи його дедалі комфортнішим. Тепер за допомогою Інтернету можна отримати доступ до різноманітного контенту, а саме освітнього, розважального, чи навіть до такого, що слугує майданчиком для просування свого бренду та продукції. Крім цього технології Інтернету речей мають потенціал до запровадження нових інноваційних процесів у таких сферах, як охорона здоров'я, транспортна галузь, та технологія розумного міста. Як видно такі інноваційні речі мають багато переваг у продуктивності, розвитку та якості життя людей.

З огляду на все це для людей завжди в пріоритеті є особиста інформаційна та фінансова безпека. Адже люди перейшли на електрону систему зберігання своїх активів, тому важливою функцією є забезпечення безпечного використання даних користувача. З урахуванням перелічених факторів та переваг, можна зробити висновок, що вживання надійних заходів та активне виявлення загроз,

зменшує ризики і дозволяє ефективно захистити систему, а отже дане дослідження є актуальним.

Wazuh є потужним інструментом, який виявляє та пом'якшує ризики в інформаційній безпеці. Він входить до сімейства SIEM із системою IDS, що забезпечує комплексне вирішення проблем безпеки. Однією з переваг даної системи є її відкритий код та активна підтримка персоналу, що робить його популярним серед інших платформ, які спеціалізуються в цій царині. В цілому Wazuh можна вважати надійним інструментом у боротьбі із шкідливим програмним забезпеченням, однак організації мають знати про усі його тонкощі задля ефективного використання на практиці.

Використання програми Wazuh охоплює широкий спектр галузей, включаючи:

- інформаційні технології: платформа застосовується для того щоб ефективно реагувати на інциденти;
- фінансові установи: щоб захистити конфіденційну інформацію клієнтів та вберегти їх від шахрайських дій;
- організації охорони здоров'я: для запобігання витоку конфіденційної інформації, а також збереженню цілісності медичних показників та записів;
- Інтернет магазини: щоб вберегти клієнтів від несанкціонованого доступу до їхніх фінансових та облікових даних;
- державні установи: для посиленого контролю та захисту критичної інфраструктури від можливих кіберзагроз та інших чинників.

Зважаючи на вищезазначене Wazuh може бути інтегрованим у будь яку структуру організації, допомагаючи їм підвищити рівень безпеки та зменшити можливі ризики, що у свою чергу підвищить їхню популярність.

Мета і завдання дослідження. Метою роботи є виявлення шкідливого програмного забезпечення за допомогою Wazuh.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- дослідити роботу платформи Wazuh;
- зробити моніторинг системи;

− ефективно виявити та зреагувати на загрози шкідливого забезпечення;

− зменшити ризики, що можуть виникнути внаслідок не вчасного реагування на інциденти;

− оптимізувати рішення для захисту цифрових активів.

Об’єкт дослідження: процеси виявлення зловмисного програмне забезпечення в режимі реального.

Предметом дослідження:методи та алгоритми виявлення зловмисного програмне забезпечення в режимі реального на основі програми Wazuh.

Наукова новизна одержаних результатів. Удосконалено алгоритм виявлення шкідливого програмного забезпечення за допомогою платформи Wazuhта його агентів.

Практичне значення отриманих результатів. Розгорнуто головні служби платформи Wazuh, до складу яких входить менеджер, дашборд та індексер, а також встановлено агента на цільову систему, який збиратиме та аналізуватиме дані що містяться в середині даної системи та виявлятиме будь які підозрілі події,що виникатимуть в ній.

Публікації та апробація КР.

1. Фляшко Н.Р., Яцків В.В. Алгоритми виявлення шкідливого програмного забезпечення за допомогою Wazuh. ЗБІРНИК НАУКОВИХ ПРАЦЬ за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2024» , Хмельницький, 2024. – С.508-512.

2. Фляшко Н., Контроль кінцевих точок з використанням агента Wazuh. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2024), Тернопіль, 2024. – С. 18-22.

1.АНАЛІЗ СИСТЕМ ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ ВТОРГНЕНЬ

1.1 Аналіз впливу шкідливого програмного забезпечення

Зловмисне програмне забезпечення стало все більше загрожувати безпеці цифрових систем. Провідні експерти з галузі кібербезпеки припускають що кількість таких атак зросте, тому критично важливим фактором для будь-якої за розміром організації є покращити модель безпеки, що відповідатиме на сучасні кіберзагрози. Згідно звіту про стан кібербезпеки, який розробила асоціація аудиту та контролю інформаційних систем ISACA було виявлено, що 69%кібер експертів вважають що частка кіберфахівців в тій чи іншій організації є недостатньою. Тому така нестача кваліфікованих робітників, які несуть відповідальність за безпеку збільшуватиме можливі ризики атак зловмисного забезпечення на підприємство.

Тенденції щодо розвитку шкідливого програмного забезпечення є динамічними. Дослідження показують, що обсяг потенційно шкідливих програм становить близько 1,2 мільярда [1], що проілюстровано на рисунку 1.1.

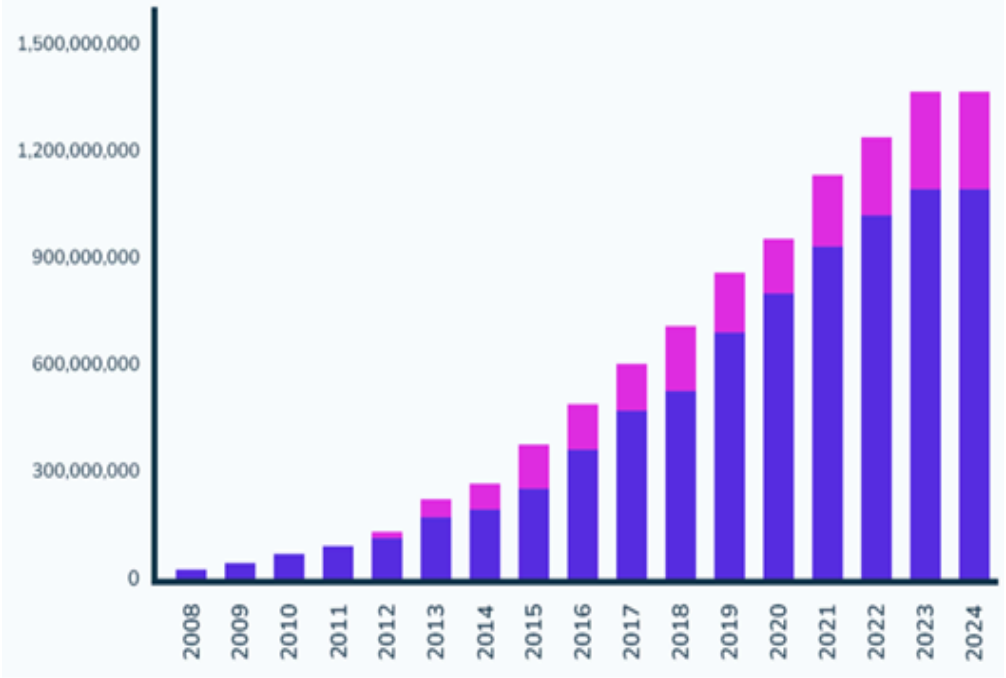


Рисунок 1.1 – Потенційно шкідливих програми

Активна конфронтація зловмисного програмного забезпечення сугубо впливає на різні галузеві сектори. Так станом на 2022 рік було зафіксовано, що найбільше атак зазнав фінансовий сектор, який в порівняні із минулорічними показниками суттєво піднявся в порівняні з іншими галузями. Діаграму атак на галузеву інфраструктуру наведено на рисунку 1.2 [1].

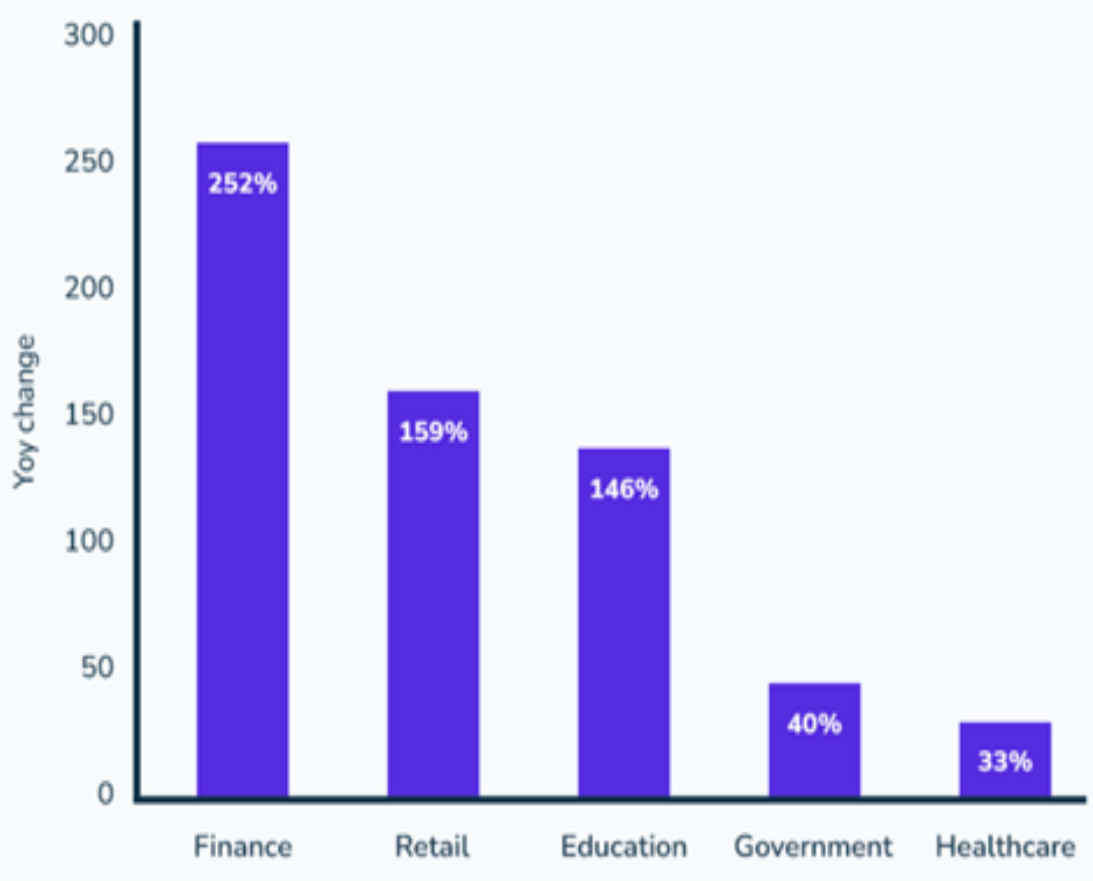


Рисунок 1.2 – Діаграма атак на галузеві сектори

Варто зазначити, що домінуючим видом атак в цифрому середовищі є програми-вимагачі. У 2022 році в усьому світі було встановлено близько 500 мільйонів випадків поширення цього шкідливого забезпечення [1]. На зараз існує багато видів програм-вимачів, але найпоширенішими є Magniber, Lockbit, Hive і BlackCat. На малюнку 1.3 зображено різноманітні моделі програм-вимачів.

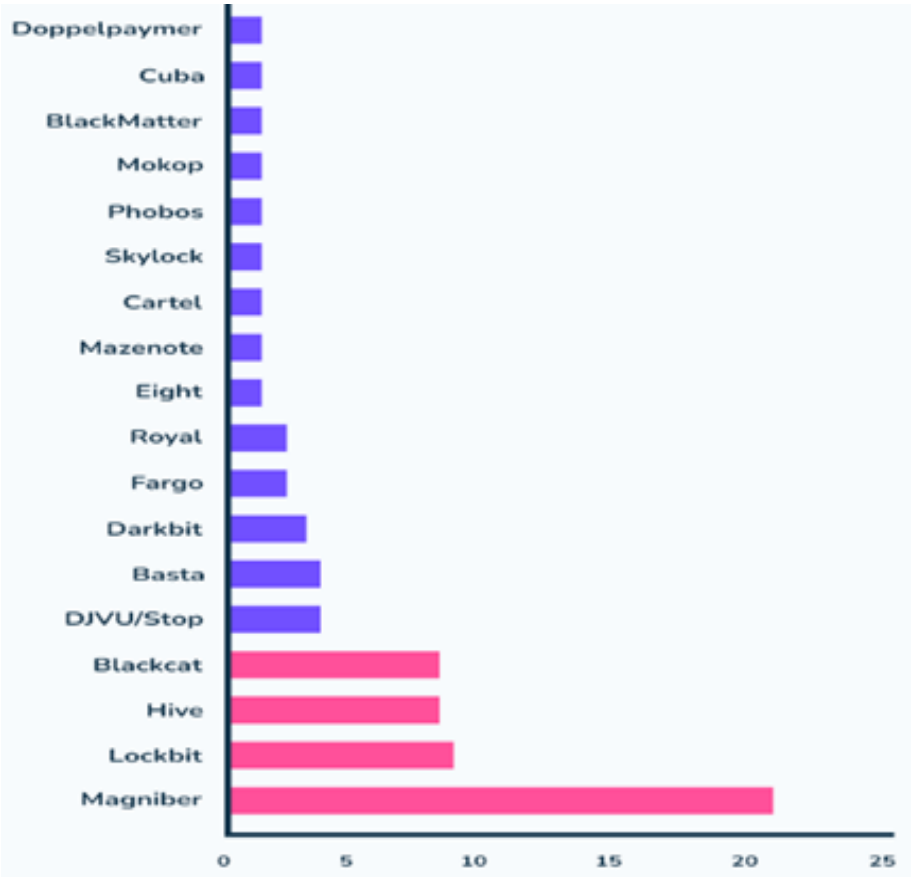


Рисунок 1.3 – Перелік моделей Ransomvare

Опрацьовані статистичні дані щодо шкідливого програмного забезпечення свідчать про те, що середовище кібербезпеки постійно змінюється. Хоча традиційні форми шкідливого програмного забезпечення, такі як програми-вимагачі, продовжують становити значну загрозу, з'являються нові види шкідливого програмного забезпечення, такі як крипто-джекинг і кампанії, спрямовані на IoT. Крім того, зростає використання штучного інтелекту для створення шкідливих скриптів.

Ці події підкреслюють важливість постійного ознайомлення з найкращими практиками кібербезпеки. Вкрай важливо, щоб отримані навички та досвід відповідали можливостям виявлення шкідливого програмного забезпечення та аналізу загроз, які потрібні організаціям для ефективної боротьби з ними.

1.2 Система виявлення вторгнень

Система IDS – це програмне забезпечення або програмний продукт, робота якого націлена виявлення шкідливих дій та порушень стосовно політик безпеки. Якщо ж буде зафіксовано, що відбулася будь яка ненормативна діяльність, то тоді адміністратор отримає повідомлення про спробу незаконного входу в його систему. Будь який несанкціонований доступ до мережевих ресурсів користувача можна сприймати як вторгнення [2]. Дана система працює за такими алгоритмами:

- відстеження активного трафіку;
- аналіз вразливих зон системи;
- перевірка цілісної структури файлів;
- співставлення відомих атак із конкретним шаблоном;
- автоматично визначає нещодавно знайдене шкідливе програмне забезпечення, яке є в Інтернеті та запобігає його проникненню в систему.

Відстеження активного мережевого трафіку системою IDS здійснюється завдяки портам TAP (Test Access Point) і SPAN (Switch Port Analyzer). Ці технологій дозволяють напряму отримувати інформацію про стан відкритих мережевий пакетів, що переміщуються по мережі. Хоча обидва порти використовуються для одної цілі, про те між ними є деякі відмінності.

Порт TAP – відносяться до апаратних пристроїв, які пасивно копіюють мережеві пакети, що проходять через нього і надсилають їх системі моніторингу. Він часто розташовується між маршрутизатором та комутатором. Ще однією із характеристик порту TAP є видимість мережевого трафіку, така функціональність дозволить точно проаналізувати вхідний та мережевий трафік для забезпечення безпеки та продуктивності системи.

З іншого боку SPAN є аналізатором портів комутатора, що створює копії мережевих пакетів, які лише частково збираються від одного визначеного порту (VLAN) до іншого конкретного порту призначеного для моніторингу.

Для ефективної роботи системи потрібно якомога глибше дослідити, з яких елементів складається цей продукт. В таблиці 1.1 наведено основні параметри IDS системи.

Таблиця 1.1 − Основні компоненти IDS системи

Будова IDS	
Підходи виявлення вторгнення	Аномальний, сигнатурний
Методи виявлення	HIDS, NIDS, Hybrids
Структурна модель	Розподілена (агент) і централізована
Відповідь системи на вторгнення	Активне та пасивне
Джерело даних	Audit trail, Network packets, SSA (kemel, services, files)
Період реагування на загрозу	Без затримок та з певним інтервалами

З вищезазначеної інформації системи виявлення вторгнень мають різні підходи, методи збору, моделі, способи реагування. На основі поданих даних сформуємо детальний аналіз цих значень.

За способом збору мережевого трафіку IDS поділяють на HIDS та NIDS.

HIDS – це важливий інструмент кібербезпеки, який призначений для виявлення шкідливого програмного забезпечення. Цей продукт встановлюється на серверних або робочих станціях [10]. Основним завданням HIDS є захист хоста мережі. Вона працює в пасивному режимі, збираючи дані про потенційні несанкціоновані дії, а також відстежуючи мережевий трафік, що перенаправляється через мережеву карту. HIDS виходить за рамки традиційного антивірусного програмного забезпечення, пропонуючи додаткові
рівні захисту, виявляючи шкідливе програмне забезпечення, яке могло прослизнути повз інші заходи безпеки в організаціях. На рисунку 1.4 подано графічну схему IDS орієнтованої на хости [2].

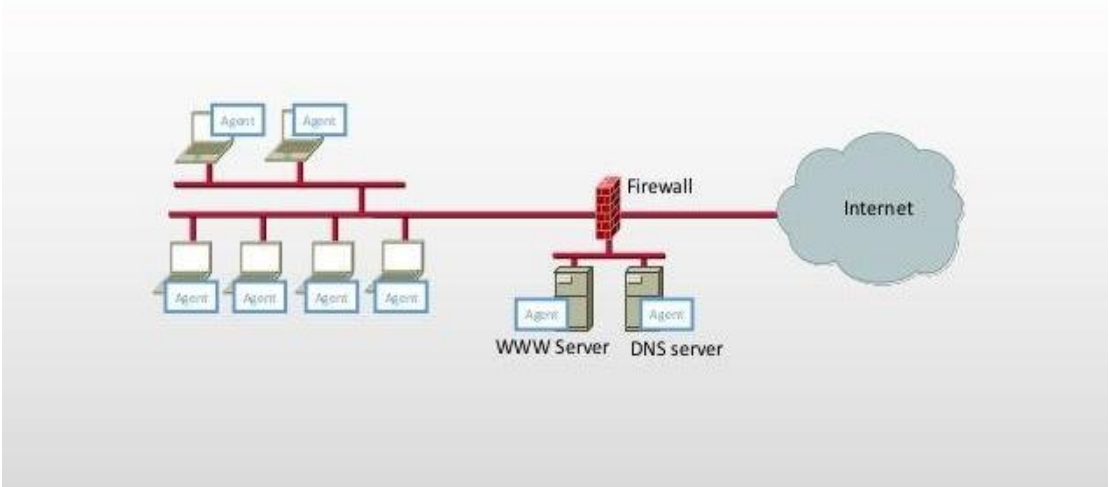


Рисунок 1.4 – Інфраструктура HIDS

Система виявлення мережевих вторгнень (NIDS) - це система моніторингу, яка спостерігає за рухом мережевих пакетів за допомогою спеціалізованих датчиків [10]. Дані, зібрані з маршрутизатора та брандмауера, передаються на центральний сервер, а потім ретранслюються на консоль для аналізу. Однак NIDS має свої недоліки, такі як нездатність розшифровувати зашифровані дані і нездатність виявляти порушення на окремих пристроях. Як наслідок, система є більш ефективною в середовищах, де трафік є незашифрованим або у звичайному текстовому форматі. На рисунку 1.5 показано інфраструктуру мережевої IDS [2].

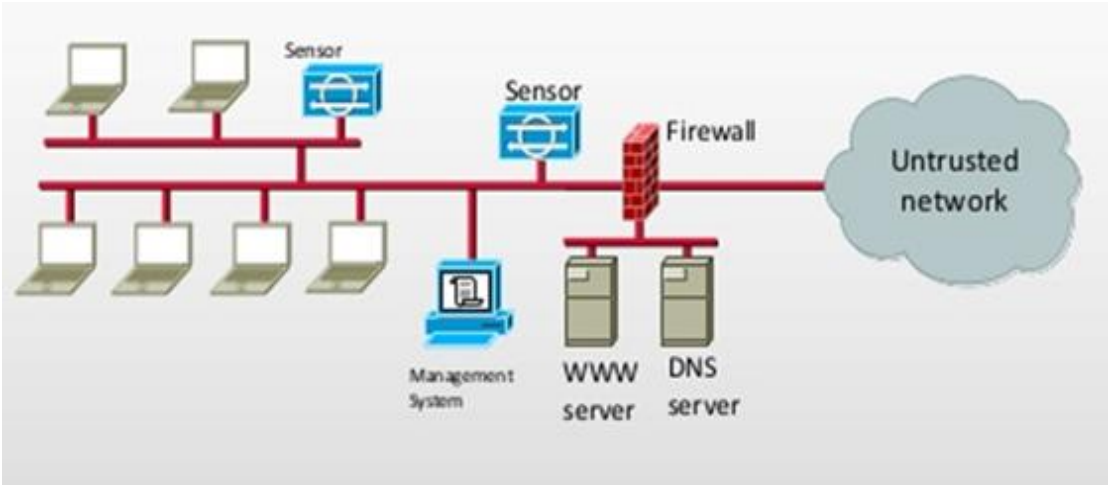


Рисунок 1.5 – Інфраструктура NIDS

Гібридна система виявлення вторгнень об'єднує різні технології IDS, що значно розширює її можливості в порівняні з іншими окремими системами [10]. Використовуючи сильні сторони кожного компонента та доповнюючи їхні слабкі, ця система досягає синергетичного ефекту, який ефективно запобігає зловмисним діям.

У сфері виявлення шкідливого програмного забезпечення одним із найпоширеніших метод на основі сигнатур та аномалій.

Метод на основі сигнатур (рисунок 1.6). Цей метод спирається на систему, яка містить бібліотеку відомих шкідливих шаблонів [3,16]. Коли сканується новий файл, якщо він збігається з будь-яким із шаблонів, що вже зберігаються в системі, він позначається як загроза. Цей процес схожий на те, як працює традиційне антивірусне програмне забезпечення - ідентифікуючи відомі віруси за їхніми сигнатурами. Для забезпечення ефективності цього методу дуже важливо регулярно оновлювати базу даних сигнатур, щоб вона могла швидко розпізнавати та усувати будь-які потенційні загрози.

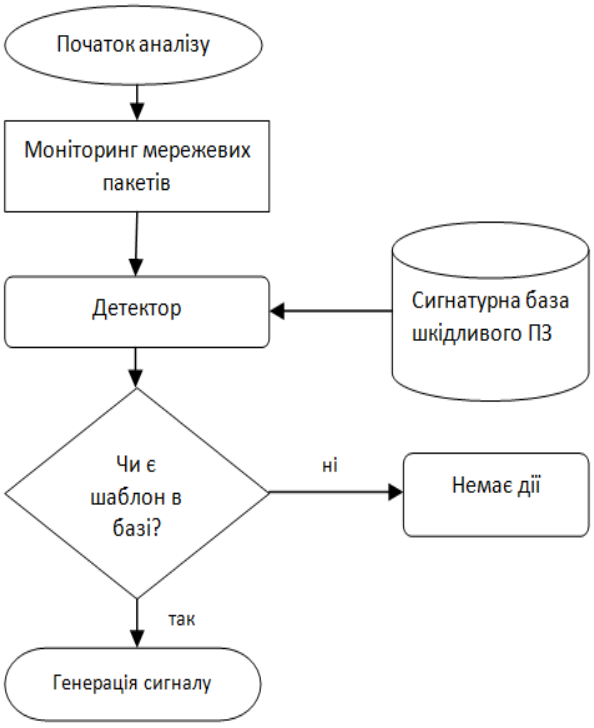


Рисунок 1.6 – Алгоритм роботи сигнатурного методу

Метод виявлення аномалій ґрунтується на виявленні незвичайних патернів у поведінці системи [3, 16, 21]. На рисунку 1.7 зображено алгоритм роботи аномалій IDS. Безперервний моніторинг активності системи та порівняння її з нормальною поведінкою дозволяє позначити будь-які відхилення як потенційні загрози [21]. Наприклад, якщо спостерігається аномальний обсяг трафіку, це може свідчити про потенційне вторгнення.

Цей підхід особливо ефективний для виявлення експлойтів нульового дня, які часто пов'язані з аномально високим рівнем трафіку. Щоб звести до мінімуму помилкові тривоги, важливо правильно відкалібрувати IDS-систему, щоб вона сповіщала лише про справжні аномалії, які можуть виникнути.

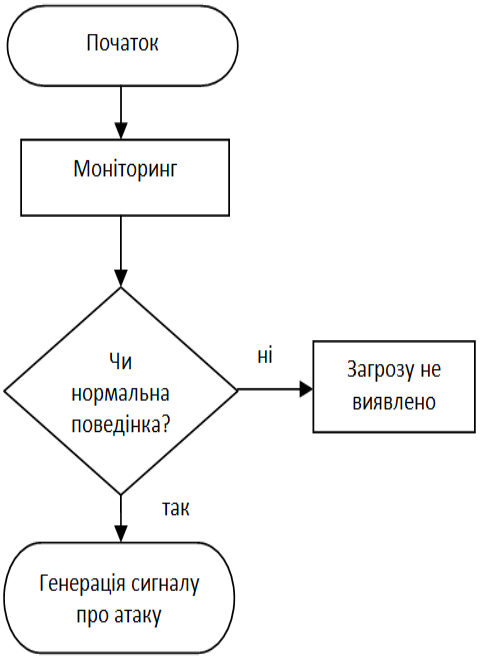


Рисунок 1.7 – Блок-схема роботи на основі аномалій

За способом повідомлення про виявлення атаки IDS поділяються на пасивні та активні. Принцип роботи пасивного IDS грунтується на реєстрі атак, які надходитимуть до адміністратора у текстовому, електронному, вспливаючому форматі. Активна IDS працює за тим же принципом, однак її особливістю є те, що вона здатна самостійно вносити корективи без активного впливу користувача.

1.3 Система запобігання вторгнень

Під технологією IPS розуміють систему, що виявляє та блокує потенційно загрози, які можуть проникнути через мережу. Таким чином, якщо аномальний трафік проник або ж була вчинена атака, то тоді ці дії розпізнаються системою і яка у свою чергу блокуватиме, виявлятиме, та вживати заходи для її успішної реалізації. Основною перевагою над системою IDS є те що вона ще до початку атаки розпізнає загрозу і усуває її, а не повідомляє, коли вже атака відбулася. Деякі системи IPS застосовують приманку щоб заманити зловмисника та не дати йому досягти поставлених задач.

На рисунку 1.8 з модельовано розташування IPS в мережі компанії [4]. Для ефективного моніторингу мережі систему IPS розміщують в середину мережевого потоку між початковим джерелом, де пакети проходять через брандмауер, тоді прямують до IPS, а потім надсилаються до точки призначення.

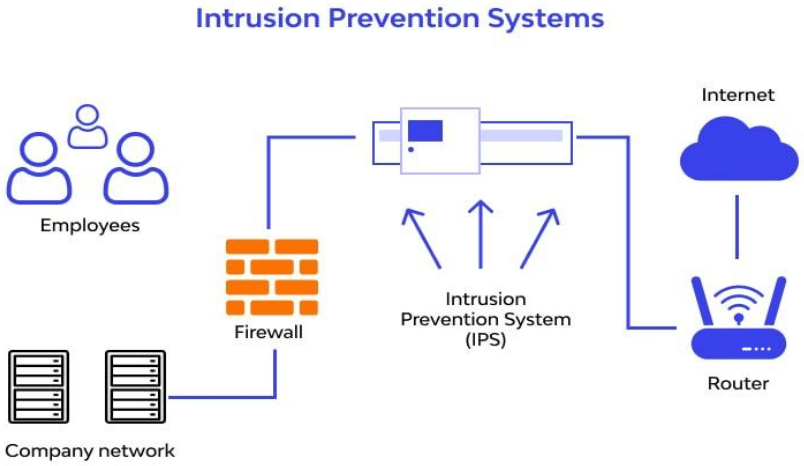


Рисунок 1.8 – Розташування IPS в мережевій інфраструктурі компанії

За методом виявлення вторгнення IPS системи засновані на:

На основі аномалій [11, 21]: ця технологія займається аналізом поведінки мережі. Тобто береться теперішній трафік, який безпосередньо порівнюється із попереднім. Якщо ж обидва трафіки відрізняються між собою, і один з них виходить за встановлені межі IPS обов’язково відреагує на це. В порівняні із

сигнатурою цей метод є більш надійним, однак інколи можуть виникати колізії пов’язані із хибними результатами.

На основі сигнатур [11]: цей спосіб націлений на виявлення експлойтів. Тобто існує база, в якій зберігаються відомі шаблони шкідливого програмного забезпечення, які відіграють важливу роль в процесі ідентифікації відомих атак. Якщо цієї атаки не має в базі, то система автоматично запише її в словник. Виявлення IPS сигнатур відбуваються за такими методами:

Виявлення експлойтів – виявлення сумісних експлойтів в мережевому трафіку із наявною базою сигнатур [11]. Або ж внесення унікальних шаблонів в бібліотеку в тому випадку, якщо не має збігу.

Виявлення вразливості – такий тип сигнатур спрямований на пошук вразливих місць, де можуть виникнути атаки [11]. Вони активно протидіють новим експлойтам підбираючи нові методи захисту. Зрештою головним недоліком зазначених сигнатур є їхнє хибне спрацювання.

Виявлення за допомогою політики безпеки: даний тип базується на правилах безпеки заздалегідь визначених компанією [11]. Якщо ж буде порушено хоча б одне з цих правил, генеруватиметься сигнал про знайдене порушення адміністратору.

Крім перелічених методик боротьби із шкідливим забезпеченням існують й певні типи IPS, що різняться за своєю функціональністю, а саме:

NIPS – відноситься до мережевих систем запобігання вторгненням. Вона призначена для захисту стратегічних параметрів мережі шляхом повного моніторингу та сканування всього мережевого трафіку [16].

HIPS – використовується для запобігання вторгнень на хост. Дана технологія виключно відповідає за аналіз вхідних та вихідних пакетів розміщених на робочих станціях. В тандемі з NIPS є невід’ємним елементом в формуванні захищеної клієнт-серверної архітектури [16].

NBA – застосовують для того, щоб проаналізувати поведінку мережі. Його робота спрямована на аналіз трафіку та виявлення нового шкідливого забезпечення або ж вразливих областей мережі [16].

WIPS – такий вид системи захищає бездротову Wi-Fi мережу, від будь яких несанкціонованих дій, блокуючи доступ до Інтернету неавторизованим абонентам [16]. В таблиці 1.2 написана порівняльна характеристика різних типів систем запобігання вторгненням.

Таблиця 1.2 – Аналіз операційних систем за наведеними параметрами

Види IPS технологій	Зони виявлення шкідливого ПЗ	Ідеї щодо вирішення	Ефективність
Мережева	Управління, передача та реалізація на рівні TCP/IP	Об'єднання підмереж у групи	Організація протоколів IDPS в під мережі
Дистанційна	Несанкціонований доступ до WLAN	Різні види WLAN	Використання IDPS в WLAN мережах
NBA	Транспортування мережевих пакетів на TCP/IP рівні	Оцінка поведінки мережі	Вживання контр заходів для визначення DoS-атак
HOST Based	Прикладний рівень TCP/IP	Впровадження нової хост інфраструктури	Аналіз трафіку на робочій станції

1.4 Впровадження програмних рішень

Сучасні системи запобігання вторгненням (IPS) та системи виявлення вторгнень (IDS) широко використовуються для виявлення шкідливого програмного забезпечення. Вибір правильної системи має вирішальне значення для ефективного вирішення проблем безпеки в мережі. Суперечки про те, що краще - IDS чи IPS, можна вирішити, інтегрувавши обидві системи в єдину систему. Для аналізу в цьому дослідженні було обрано таке ПЗ:

- OSSEC HIDS для IDS;
- Snort NIPS для IPS;
- PATP для IDPS;

– Wazuh.

OSSEC, також відома як Open Source Security, - це широко використовувана система виявлення вторгнень (HIDS) з відкритим вихідним кодом, яка пропонує аналіз журналів в режимі реального часу, перевірку цілісності файлів, виявлення руткітів і можливості активного реагування [5, 20]. Нижче наведено детальний опис роботи OSSEC.

1. Аналіз журналів: OSSEC збирає і аналізує дані журналів з різних джерел, включаючи системні журнали, журнали додатків і мережевих пристроїв [20]. Вона використовує заздалегідь визначені правила і декодери для аналізу і стандартизації повідомлень журналів для подальшого вивчення.

2. Перевірка цілісності файлів: OSSEC пильно стежить за критично важливими системними файлами і каталогами, щоб виявити будь-які несанкціоновані зміни [20]. Вона обчислює контрольні суми файлів і порівнює їх зі встановленими базовими значеннями, щоб виявити будь-які зміни або втручання.

3. Виявлення руткітів: OSSEC оснащений функціями виявлення руткітів для виявлення шкідливого програмного забезпечення, яке намагається приховати свою активність в системі [20]. Він сканує системні двійкові файли та конфігураційні файли на наявність ознак поведінки руткітів.

4. Активне реагування: OSSEC можна налаштувати на автоматичне реагування на інциденти безпеки, такі як блокування IP-адрес, деактивація облікових записів користувачів або запуск користувацьких сценаріїв на основі попередньо визначених правил [20].

5. Централізоване управління: OSSEC забезпечує централізоване управління за допомогою зручного веб-інтерфейсу або інструментів командного рядка [20]. Це дозволяє адміністраторам контролювати і обробляти оповіщення про загрози безпеки з декількох систем на консолідованій інформаційній панелі.

По суті, OSSEC є гнучким інструментом безпеки, який дозволяє організаціям посилити свій захист шляхом швидкого виявлення та усунення потенційних ризиків безпеки [20]. Високий рівень кастомізації робить його

придатним для задоволення конкретних потреб безпеки в різних середовищах. На рисунку 1.9 наведено інтерфейс OSSEC ресурсу.

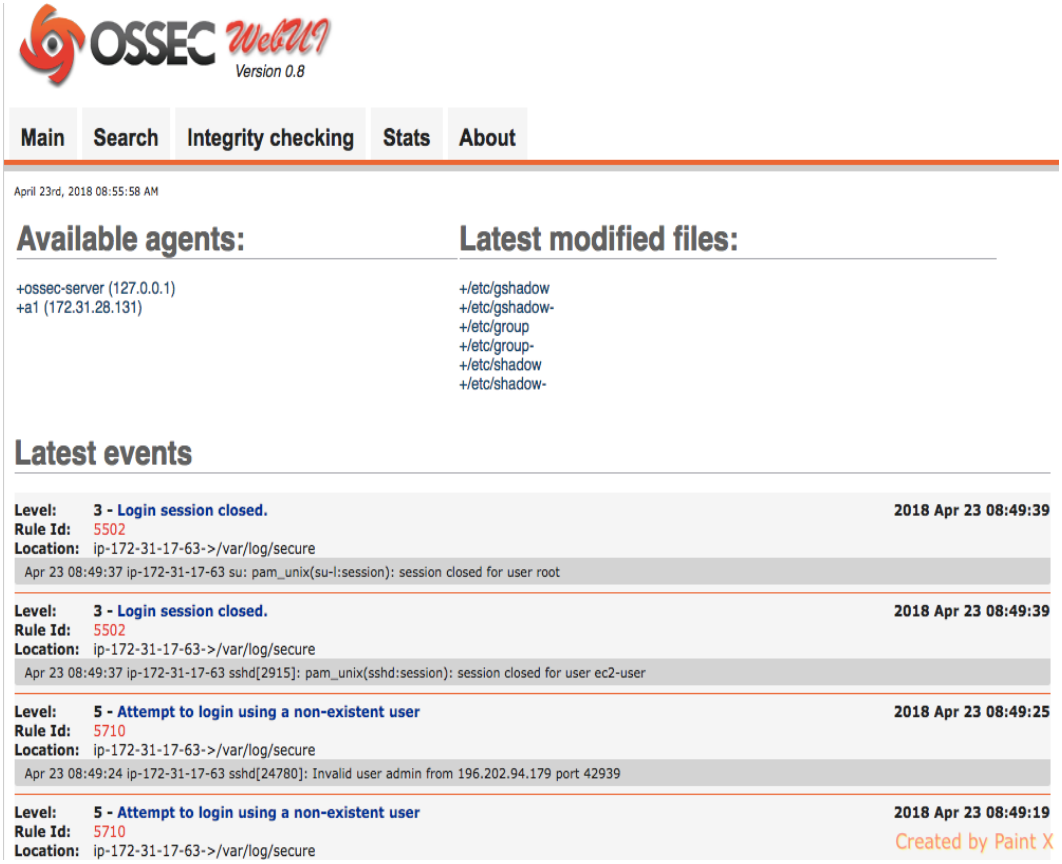


Рисунок 1.9 – Інтерфейс OSSEC ресурсу

Suricata - це сучасна система виявлення та запобігання вторгнень з відкритим вихідним кодом (IDS/IPS), яка була розроблена для контролю мережевого трафіку на предмет виявлення будь-яких зловмисних дій та надання негайних сповіщень і варіантів запобігання [18]. Нижче представлено детальний опис основних можливостей та функцій Suricata.

1. Багатопотокова структура: Suricata побудована на основі багатопотокової архітектури, що дозволяє їй ефективно аналізувати та обробляти мережевий трафік на різних процесорних ядрах. Ця здатність дозволяє Suricata управляти великими обсягами мережевого трафіку без шкоди для продуктивності.

2. Сумісність з різними протоколами: Suricata пропонує підтримку різноманітних мережевих протоколів, таких як TCP, UDP, ICMP, HTTP, DNS і TLS. Він може досліджувати як IPv4, так і IPv6 трафік, що робить його придатним для сучасних мережевих налаштувань.

3. Виявлення на основі сигнатур: Suricata використовує виявлення на основі сигнатур для виявлення розпізнаних загроз і атак на основі попередньо встановлених наборів правил. Ці набори правил налаштовуються і можуть регулярно оновлюватися, щоб бути в курсі нових загроз.

4. Виявлення на основі аномалій: Окрім виявлення на основі сигнатур, Suricata також використовує методи виявлення на основі аномалій для виявлення підозрілих дій та відхилень від нормального мережевого трафіку. Це корисно для виявлення атак нульового дня і невідомих загроз.

5. Вилучення та аналіз файлів: Suricata має можливість витягувати і ретельно аналізувати файли з мережевого трафіку, що дозволяє виявляти і знешкоджувати шкідливі файли або корисне навантаження. Процес вилучення файлів може базуватися на таких атрибутах, як тип файлу, розмір тощо.

6. Журналювання та звітування: Suricata створює детальні журнали та звіти про виявлені події, пропонуючи адміністраторам розуміння мережевих операцій та інцидентів безпеки [22]. Вона підтримує різні вихідні формати, такі як JSON, EVE JSON і unified2 для інтеграції з рішеннями SIEM [22].

7. Інтеграція з каналами розвідки загроз: Suricata може легко інтегруватися з каналами розвідки загроз, щоб покращити свої можливості виявлення та блокувати відомі шкідливі IP-адреси, домени та URL-адреси в режимі реального часу.

Таким чином, Suricata є потужним і адаптивним рішенням IDS/IPS, яке допомагає організаціям ефективно захищати свої мережі від кіберзагроз. Розширені функціональні можливості, підвищення продуктивності та гнучкість роблять його безцінним ресурсом для фахівців з мережевої безпеки. Відповідно, на рисунку 1.10 зображено інтерфейс користувача Suricata [24].

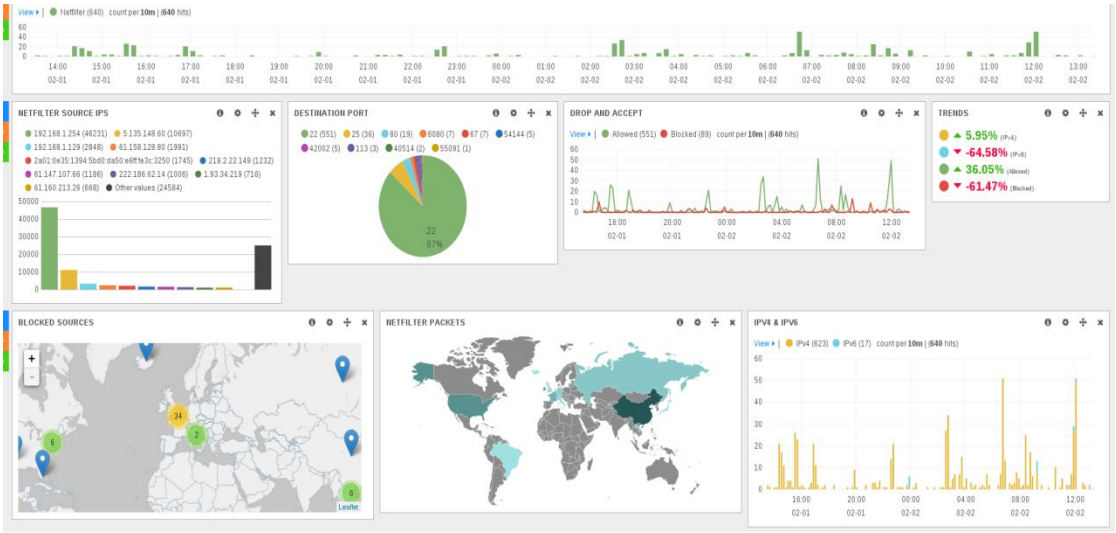


Рисунок 1.10 – Користувальницький інтерфейс Suricata

Snort, популярна система виявлення та запобігання мережевим вторгненням з відкритим вихідним кодом (NIDS/NIPS), пропонує аналіз і реєстрацію пакетів в реальному часі для IP-мереж [18, 23]. Нижче ми детально розглянемо, як працює Snort [23].

1. Захоплення пакетів: Snort перехоплює мережевий трафік, дозволяючи мережевим інтерфейсам працювати в режимі проміскуїтету, що дозволяє йому вивчати всі пакети, які проходять через мережевий сегмент.

2. Аналіз пакетів: Snort розшифровує перехоплені пакети, щоб отримати деталі, такі як IP-адреси джерела та призначення, порти, протоколи та інформацію про корисне навантаження

3. Виявлення на основі правил: Використовуючи механізм виявлення на основі правил, Snort перевіряє мережевий трафік на відповідність заздалегідь визначеному набору правил. Ці правила можна налаштувати для виявлення певних шаблонів, сигнатур або поведінки, пов'язаних з відомими загрозами

4. Сповіщення: Виявивши відповідність мережевого трафіку правилу, Snort генерує сповіщення, що містить інформацію про виявлену загрозу, включаючи тип атаки, IP-адреси джерела та призначення, а також часову мітку.

5. Ведення журналу: Snort записує всю мережеву активність і попередження в різні вихідні формати, такі як текстові файли, бази даних і

сервери syslog. Ця функція логування дозволяє адміністраторам аналізувати події безпеки для реагування на інциденти та судового розслідування.

6. (NIPS): Окрім ролі системи виявлення вторгнень (NIDS), Snort може функціонувати у вбудованому режимі як система запобігання мережевим вторгненням (NIPS). У цьому режимі Snort може активно блокувати або відкидати шкідливий трафік на основі попередньо визначених правил.

7. Сумісність з протоколами: Snort підтримує широкий спектр мережевих протоколів, включаючи TCP, UDP, ICMP, IP і протоколи прикладного рівня, такі як HTTP, FTP, SMTP і DNS.

Загалом, Snort - це надійний і адаптивний інструмент безпеки, який допомагає організаціям контролювати і захищати свої мережі від різних загроз, таких як шкідливе програмне забезпечення, експлойти і підозріла активність [23]. Механізм виявлення на основі правил, набори правил, що налаштовуються, і функції ведення журналів роблять його безцінним інструментом для моніторингу мережевої безпеки та реагування на інциденти.

В якості програмного ресурсу було вибрано платформу моніторингу Wazuh. завдяки своїй здатності забезпечити комплексний підхід до забезпечення безперебійної роботи користувачів.

Ця платформа легко адаптується і може бути легко розгорнута в різних середовищах. Проактивно реагуючи на потенційні інциденти, компанії можуть в кінцевому підсумку заощадити на витратах, пов'язаних з ризиками.

Використання каналів зв'язку дозволяє ефективно поширювати точні сповіщення про потенційні загрози. В результаті організації можуть захистити свої системи як від нових, так і від ідентифікованих загроз шкідливого програмного забезпечення, отримуючи постійну допомогу від спільноти розробників у разі виникнення будь-яких проблем. Система оснащена засобами для виявлення та запобігання вторгненням, а також центральним компонентом, який збирає важливі дані та показники продуктивності системи. Таким чином, Wazuh ефективно вирішує наступні завдання:

- збір інформації про стан системних log-файлів, ототожнивши його із справжнім вмістом log-файлу;
- моніторинг цілісності файлів: агент перевіряє вміст файлу та визначає чи не були модифіковані його дані;
- виявлення вторгнення: аналізує поведінку мережевого трафіку, системні події, та інші показники;
- активне реагування: автоматично реагує на потенційні інциденти, блокуючи шкідливі IP адреси, створюючи ізоляцію системи, запроваджуючи сценарії по стримуванні загрози;
- моніторинг відповідності: допомагає організаціям діяти згідно чино встановлених норм і стандартів, що контролюють безпеку.

Щоб Wazuh ефективно функціонував як система виявлення вторгнень (IDS) та оперативно сповіщав користувача про виявлені вразливості, рекомендується налаштувати параметри сповіщень відповідно до уподобань користувача для зручності доступу.

Щоб налаштувати Wazuh для системи запобігання вторгненням (IPS), необхідно використовувати ряд додаткових програм Active Response. Ці програми призначені для активації тригерів, які активно збирають дані від серверного середовища до кінцевих точок мережі. Крім того, ці програми мають можливість:

- блокування підозріло активного IP адресу;
- автоматичне закриття мережі;
- запровадження нових правил конфігурації брандмауера;
- знешкодження шкідливих процесів;
- зручне донесення інформації.

Підсумовуючи, Suricata, Snort і Wazuh - це першокласні рішення для забезпечення безпеки з різноманітними функціями для виявлення та моніторингу загроз.

Suricata виділяється своїми винятковими можливостями виявлення та запобігання мережевим вторгненням, що робить його кращим варіантом для

організацій, які прагнуть захистити свої мережеві налаштування [18]. Масштабованість і адаптивність роблять його придатним для використання в різних середовищах, від невеликих установ до великих підприємств.

Snort, з іншого боку, є добре зарекомендованою системою виявлення вторгнень з відкритим вихідним кодом, відомою своїм методом виявлення на основі сигнатур і великими наборами правил. Це робить її надійним вибором для моніторингу мережевої безпеки.

Wazuh слугує комплексною платформою для моніторингу безпеки, що пропонує широкий спектр функцій, таких як виявлення вторгнень, аналіз журналів, виявлення вразливостей та управління відповідністю вимогам. Вона забезпечує виявлення загроз у режимі реального часу та реагування на них, що робить її корисним інструментом для організацій, які прагнуть посилити свої заходи безпеки.

Використання цих систем варіюється залежно від унікальних потреб і вимог кожної організації. Suricata, Snort і Wazuh широко використовуються в різних галузях, від малих підприємств до великих корпорацій, з метою моніторингу та захисту мережевої інфраструктури від кіберзагроз. По суті, ці рішення вважаються ефективними інструментами у сфері кібербезпеки, які широко застосовуються організаціями для посилення їхнього захисту [19].

Загалом, система Wazuh є надзвичайно зручним у використанні інструментом, який забезпечує надійність, точність і гнучкість, що робить її придатною для будь-якої організації, яка прагне посилити свої заходи безпеки [18]. Завдяки можливості об'єднання декількох функцій в систему Wazuh, вона може легко інтегруватися як з системами запобігання вторгненням (IPS), так і з системами виявлення вторгнень (IDS), що робить її потужним і адаптивним рішенням для посилення безпеки.

Однією з помітних тенденцій у сфері кібербезпеки є дедалі ширше використання штучного інтелекту (ШІ) [18]. ШІ можна використовувати для автоматизації процесів тестування на проникнення, що в кінцевому підсумку підвищує ефективність.

Крім того, в кібербезпеці все більше уваги приділяється використанню хмарних обчислень. З впровадженням технологій хмарних обчислень фахівці з кібербезпеки тепер можуть проводити тестування на проникнення в більших масштабах, які раніше були недосяжними.

Очевидно, що Kali Linux залишиться важливим інструментом в майбутньому кібербезпеки, адаптуючись до нових тенденцій і зберігаючи свою актуальність і ефективність в області тестування на проникнення.

2 СТРУКТУРА ДОСЛІДЖУВАНОГО СЕРЕДОВИЩА WAZUH

2.1 Архітектура платформи Wazuh

Wazuh - це універсальне рішення, яке об'єднує технології системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS). Платформа побудована на основі функцій розширеного виявлення та реагування (XDR) і управління інформацією та подіями безпеки (SIEM) [22]. Вона здатна оцінювати дії поточних процесів у режимі реального часу. Крім того, вона може перевіряти цілісність даних, виявляти та зупиняти підозрілу поведінку, а також перевіряти вхідний мережевий трафік на відповідність попередньо визначеним критеріям.

Основним компонентом системи Wazuh є агент. Однак для того, щоб встановити з'єднання, необхідно спочатку встановити менеджер Wazuh. Функціонуюча система Wazuh призначена для ефективної роботи в різних середовищах, таких як хмарні платформи (AWS, Microsoft Azure, GitHub), різноманітні операційні системи (Linux, Mac, Windows), віртуальні середовища тощо. Ця інноваційна технологія в першу чергу відповідає за збір, аналіз та представлення даних на дашборді Kibana.

Отже, після ретельних роздумів, віртуальну машину під назвою VirtualBox було обрано як робоче середовище для запуску операційної системи Kali разом з вразливим Metasploit. Завантажувальний файл з розширенням .ova буде розміщено у віртуальному середовищі, і ми будемо дотримуватися інструкцій, наведених у документації для розробників Wazuh. Повну розбивку цієї установки можна знайти на рисунку 2.1, який ілюструє ключові елементи архітектури досліджуваної платформи.

Ця система призначена для ефективної роботи як на стороні сервера, так і на стороні агента. Агент відповідає за збір даних, які потім будуть проаналізовані та передані іншим суб'єктам моніторингу. Ці дані обробляються аналітичним механізмом, здатним оперативно виявляти вразливості або потенційні загрози. Окрім аналітичного механізму, платформа також включає в себе інструмент

Kibana, який пропонує можливості пошуку та візуалізації. Ці компоненти разом забезпечують розширену функціональність, дозволяючи користувачам ефективно отримувати доступ до інформації за допомогою методів сортування та візуалізувати важливі дані за допомогою зручного інтерфейсу. На рисунку 2.1 показано архітектуру Wazuh платформи [17].

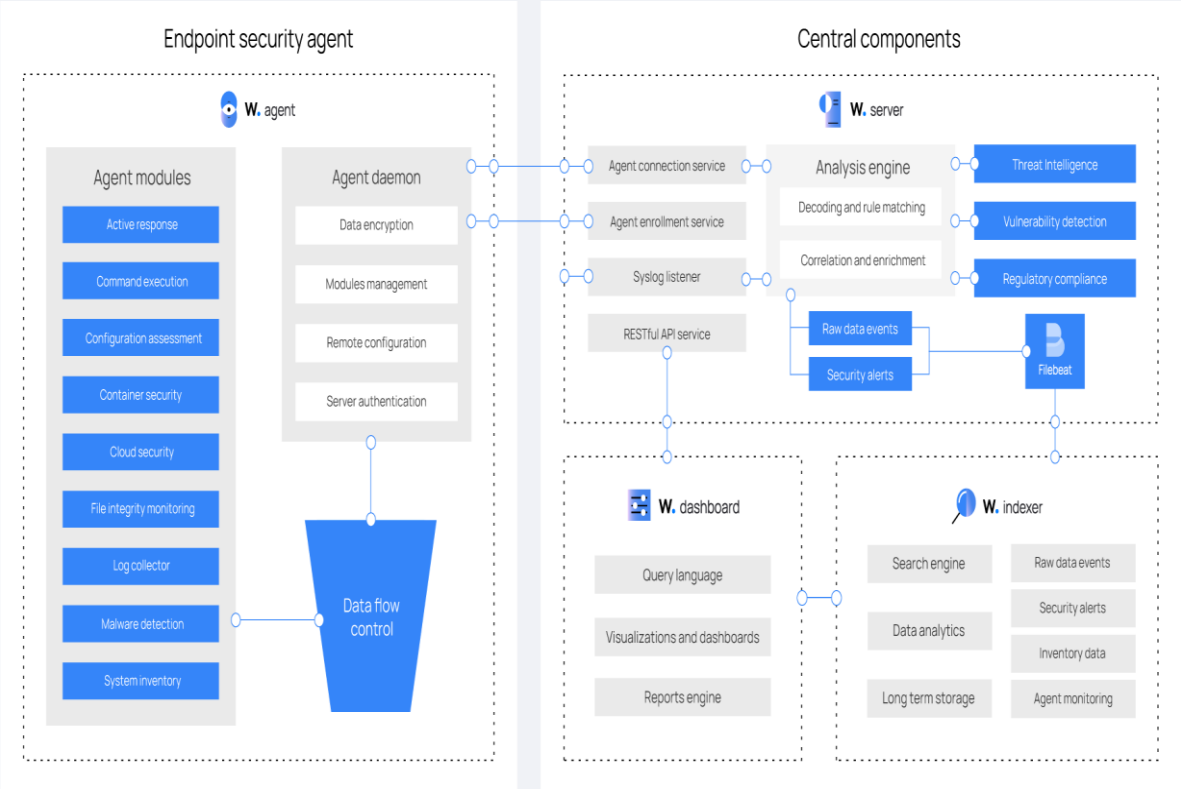


Рисунок 2.1 - Архітектурна концепція досліджуваної Wazuh платформи

Розглядаючи структурні елементи системи, стає очевидним, що агент Wazuh стратегічно розгортається на визначених кінцевих точках і доповнюється трьома адаптивними модулями: індексатором, сервером та інформаційною панеллю. Нижче ви знайдете детальний опис функціональності кожного компонента.

Wazuh Indexer відіграє важливу роль в екосистемі Wazuh, проактивно шукаючи та аналізуючи сповіщення, що надходять від сервера Wazuh. Він може бути адаптований до різних вимог, як для одного пристрою, так і для декількох вузлів, гарантуючи масштабованість і легкий доступ.

Індексатор зберігає дані формату JSON. Завдяки такому способу організації ви можете пов'язувати групи ідентифікаторів, атрибутів або інших характеристик, таких як текст, цифри, логічні твердження та додаткові вхідні дані, або інші значення.

Основною властивістю індексатора є збір взаємопов’язаних документів. Ці документи розподіляються між сегментами тим самим створюючи окремі вузли. Такий підхід сприятиме виділенню додаткового значення, що продукуватиме надлишковість. Зрештою, це призводить до покращення продуктивності та збільшення пропускної здатності запитів у системі.

Система індексації складається з чотирьох різних категорій: Сповіщення Wazuh, архіви, моніторинг та статистика. Кожен підтип призначений для управління певними сферами безпеки, про що свідчать їхні назви.

На рисунку 2.2 бачимо програмний код запиту, що отримує індексатор від сервера до вузла [25].


```
{
 "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"
   ]
...
```


Рисунок 2.2 – Програмний код запиту

Індексатор Wazuh - це група з декількох індексів, кожен з яких представляє різні джерела або категорії документів про події. У цьому кластері, який у нашому сценарії є індексатором Wazuh, є вузли, що містять фрагменти документів про події, відомі як шарди. Використовуючи шарди, система може підтримувати постійну доступність і безперебійну роботу. Це досягається завдяки надлишковому зберіганню первинних і вторинних даних на кожному вузлі. У випадку шардів реплік первинні екземпляри можуть бути розподілені між кількома вузлами, що забезпечує доступ до даних, навіть якщо не всі вузли працюють одночасно.

Даний продукт є ефективним у ідентифікації шкідливого програмного забезпечення. Реагування на будь які інциденти відбувається у швидкому режимі без великих затримок. Така обробка даних гарантуватиме якісну оцінку у введені журналювання подій в операційних системах. На рисунку 2.3 проілюстровано структурну схему Wazuh indexer.

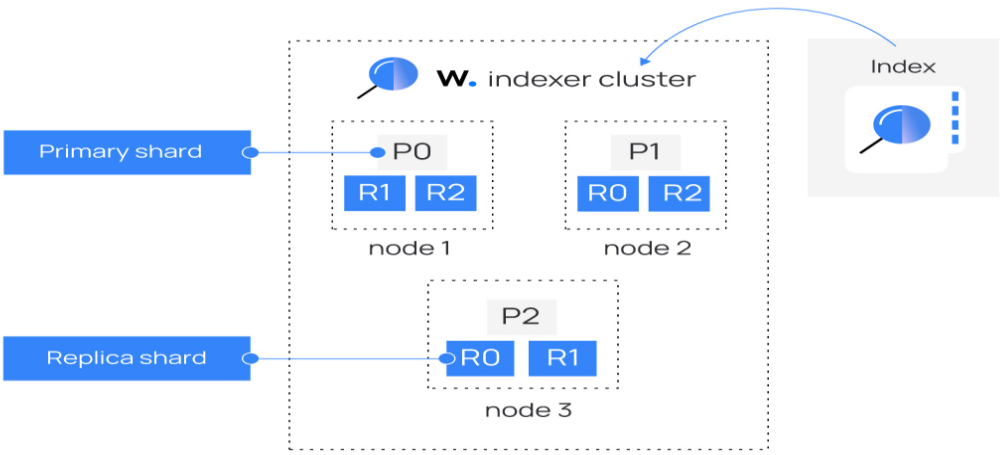


Рисунок 2.3 – Будова Wazuh indexer

Wazuh сервер це досить дієва платформа яка вирішує ряд головних завдань зорієнтованих на отримані даних від агентів. Це дозволить як найкраще виявити та запобігти ймовірним загрозам. Його інтерфейс можна розгорнути на будь якій іншій робочій області на яку не встановлено сервер. Таким чином в онлайн

режимі оператор вузла отримає інформацію про стан програмного забезпечення на віддаленій дистанції.

В сам сервер вбудовані джерела, що займаються аналізом загроз. Наприклад структура MITER ATT&CK попереджає про відхилення мережевого компоненту від встановлених нормативних вимог. До нормативних вимог можна віднести HIPPA,GDPR, PCI DSS, та CIS, NIST [15].

Однією з переваг сервера є те, що він лояльний до тої чи іншої зовнішньої системи. Він вдало підійде для систем управління задачами та обслуговуванням клієнтів (ServiceNow, Jira та PagerDuty), а також можливість взаємодіяти із комунікаційною платформою SLACK.

Архітектура сервера має різнобічну будову. Кожний компонент є взаємозалежним один від одного. Центральним елементом архітектури сервера вважається Wazuh RESTful API. Як відомо сервер відповідає за Зазвичай підходящим робочим середовищем для оперативної роботи сервера вважається операційна система Linux, а також сервер можна запустити і на альтернативних ресурсах на кшталт віртуальної та автономної фізичної машини, хмарному екземпляру та на докер контейнері. На рисунку 2.4 наведено архітектуру Wazuh сервера із його службами [26].

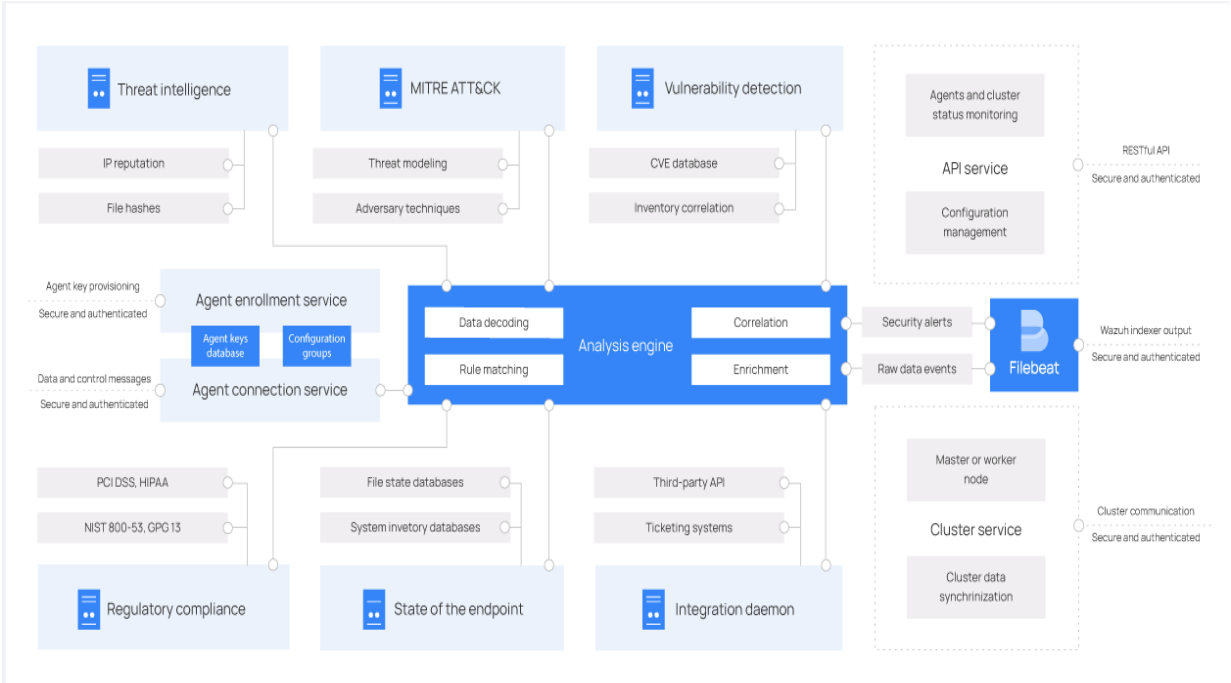


Рисунок 2.4 – Архітектура Wazuh сервера із його конфігуруючими службами

Із поданої схеми видно, що серверна частина системи Wazuh складається із багатьох служб, які мають різноманітну функціональність, а саме внесення в базу незнайомих агентів, зіставлення агентів із наявними та встановлення захищено з'єднання між агентом та сервером Wazuh. Щоб зрозуміти їхню роботу варто ретельно дослідити кожну елементну одиницю окремо, а саме реєстрацію агентів, підключення агентів, аналіз Wazuh, API RESTful, кластер Wazuh, Wazuh Filebeat.

1.Реєстрація агентів – це служба, яка полегшує роботу щодо додавання нових агентів до менеджера Wazuh. Кожному агенту належатиме свій унікальний ключ аутентифікації. Процес виконання служби базується на роботі TLS/SLL сертифікатів аутентифікації. Такий централізований механізм дозволить ефективно присвоїти та налаштувати агент на правильну роботу в інфраструктурі безпеки, забезпечуючи комплексне покриття мережі та ефективну її реалізацію.

2.Підключення агентів – це служба, яка встановлює зв'язок між агентом та менеджером Wazuh. Таке сполучення проходить лише в зашифрованому вигляді. В такому випадку зіставлятимуться ключі та перевірятимуться кожні агенти на їх достовірність. Крім цих аспектів, підключення нових параметрів агента можна виконати дистанційно, це підкреслює їх централізованість

3. Аналізатор Wazuh – це механізм, який відповідає за введення цілісного аналізу інформації (журнали подій операційної системи, журнали SSH та веб-сервера, тощо). Процес аналізу відбувається шляхом витягування важливих елементів, як IP адрес, ідентифікатор потоку даних, або назва користувача. Далі це все порівнюється із нормативними вимогами. Якщо вони не відповідатимуть цим правилам, то механізм акумулюватиме контрзаходи щоб захистити систему користувача.

4. API RESTful – це служба, що відображає інтерфейс користувача Wazuh. В загальному, вона застосовується для управління агентами та серверами, моніторингу робочої області, запитування станів взаємозалежних вузлів,

регулювання та керування вхідними даними, що обробляються декодером, а також встановлює правила та визначає стан машин, що досліджуються.

5. Кластер Wazuh – це програмний компонент, який забезпечує горизонтальну кластеризацію менеджера Wazuh. Розгортання кластеру в організаційних аспектах сприятиме високій продуктивності, масштабованості та відмовостійкості. Між іншим такий тип конфігурації слугує комунікаційним інструментом між серверами та компактним методом з'єднання між ними.

6. Wazuh Filebeat – це інструмент розроблений Elastic, який займається збором файлів з журналів різних джерел, обробці та надсиланню повідомлення до індексатора Wazuh. Він зчитує запущені файли системою в онлайн режимі. Окрім цього зазначений інструмент збалансовує трафік при під'єднанні кількох індексаторів до кластера.

Не менш важливим компонентом системи є інформаційна панель Wazuh. Дашборд призначений для візуального представлення досліджуваних агентів та їхнього статусного плану в моніторинговому порядку. Графічний дизайн забезпечуватиме зручність при його використанні на практиці. На ньому розміщуються усі важливі показники. На рисунку 2.5 показано графічний інтерфейс користувача. В графічній оболонці користувач зможе створювати різноманітні таблиці, діаграми, а також власні звіти завдяки вбудованим інструментам [27].

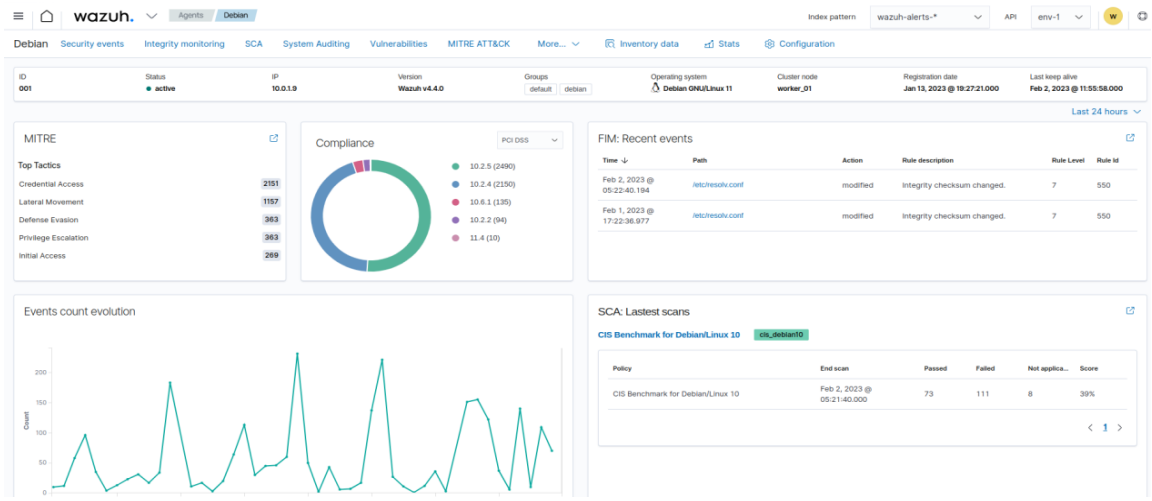


Рисунок 2.5 – Панель керування

Варто зазначити, графічний ресурс наділений різноманітними інформаційним панелями, кожна з яких підпорядковуватиметься конкретній нормативній вимозі наприклад стандарту даних платіжних карт (PCI DSS), загального регламенту про захист даних (GDPR), а також NIST 800-53 документ, який встановлює стандарти та рекомендації для кібербезпеки в урядових системах інформаційної безпеки.

Візуалізація моніторингових процесів забезпечується агентами. Як правило вони несуть інформацію про стан, що перебувають в активному режимі. Користувачі можуть визначити які агенти є доступними, як журнали опрацьовані, та які файлі є цілісними. На рисунку 2.6 проілюстровано нормативні правила панелі управління Wazuh [28].

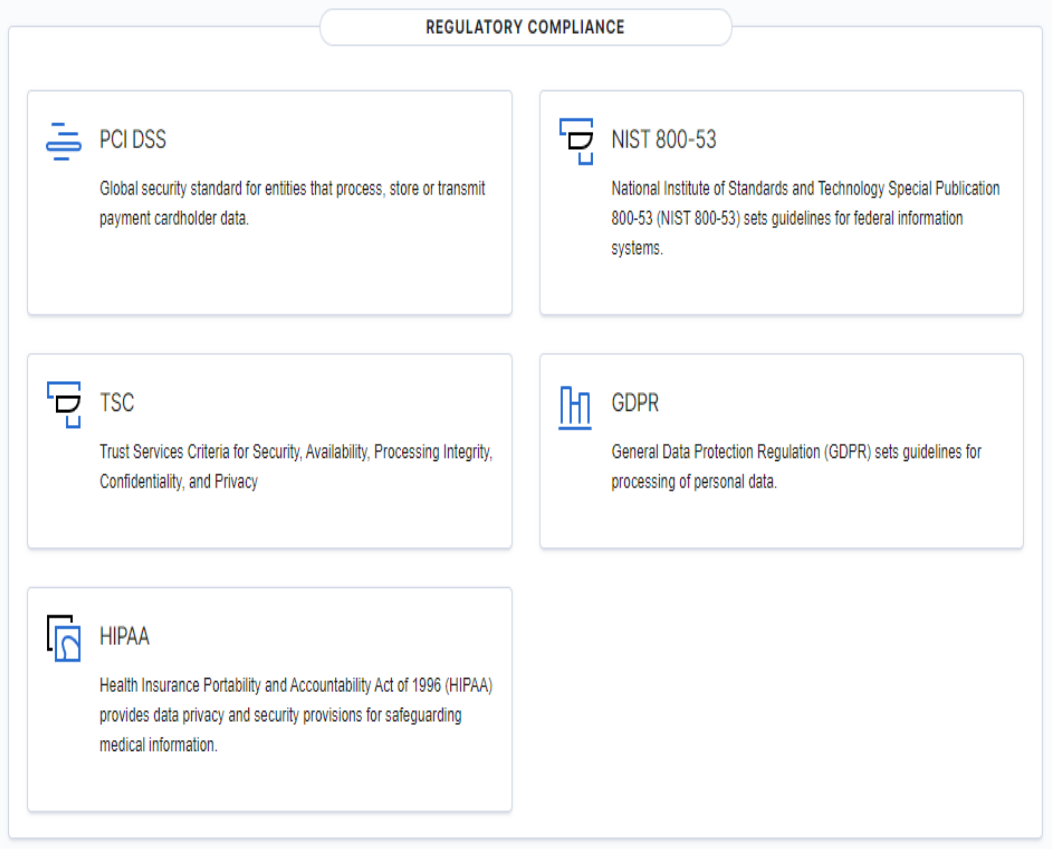


Рисунок 2.6 – Нормативні правила дашборду

Адміністрування панелі Wazuh включає набір правил, які обробляють та виявляють за допомогою декодера, який дає відповідь на запитання: чи знайдено

якесь не зі стикування із визначеними правилами? Ця функція є корисною в тому випадку, коли користувача захоче запровадити нові правила та протестувати завдяки змодельованому коду. На рисунку 2.7 показано адміністративну панель розробника [30].

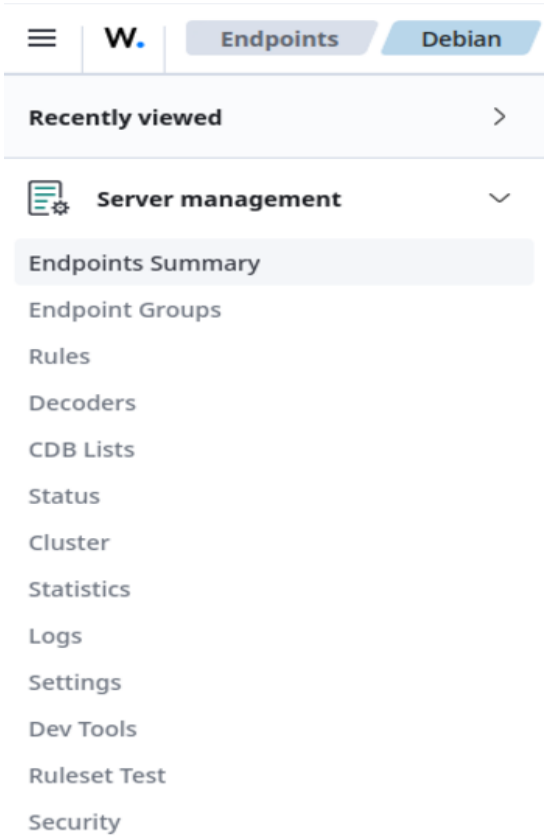


Рисунок 2.7 – Адміністративна панель розробника

Тим не менш, основним каталізатором фреймворку Wazuh є агент. Цей агент може бути встановлений на різні операційні системи, такі як Windows, Linux, macOS та докер-контейнери. Його адаптивність дозволяє йому працювати на різних пристроях, таких як ноутбуки, сервери, віртуальні машини та інші робочі простори. Діючи як проактивний захисний механізм, агент захищає систему, зупиняючи потенційні атаки до того, як вони зможуть завдати шкоди, а також ідентифікує та знешкоджує шкідливе програмне забезпечення за допомогою інноваційних методів IPS та IDS. Крім того, для підтримки безпечного зв'язку канали, що з'єднують агента і сервер, шифруються для додаткового захисту.

Аналізуючи структурну ієрархію агента Wazuh можна ствердити, що він складається з численних модулів. Кожний елемент модуля заточений на виконання завдань в тій робочій області за яку він відповідає. Вони реалізують багато функцій, серед них можна виділити наступні: моніторинг файлів, сканування log-файлів, ініціювання даних інвентаризації, зчитування конфігураційні файлів. Користувачам надається право на редагування конфігураційного файлу агента, коли виникне потреба налаштування агенту для реалізацій конкретних цілей. На рисунку 2.8 наведено візуальне представлення архітектури агента Wazuh [29].

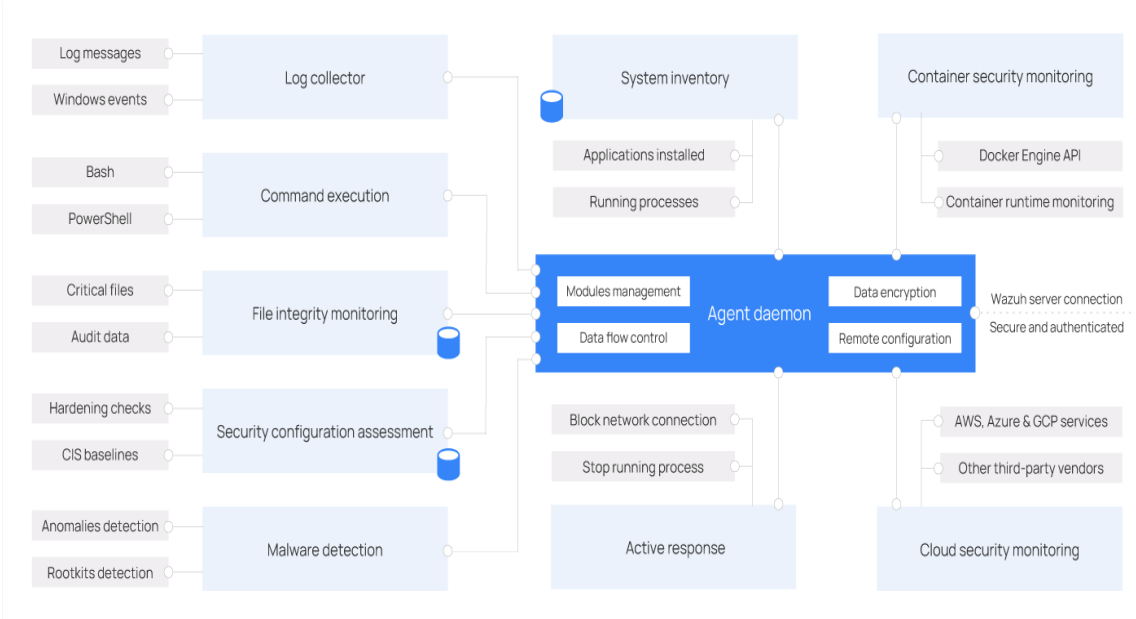


Рисунок 2.8 – Візуальне представлення архітектури агента Wazuh

Модулі агента виконують різні завдання. В залежності від задач агенти налаштовують так, щоб вони вирішували лише ті завдання, з якими вони можуть справитися. Нижче деталізовано описано роль кожного агента в системі та основне їх призначення. До ключових модулів агента можна віднести модуль збирача журналів, командна реалізація, моніторинг та цілісність файлів, оцінку безпеки конфігурації, агент інвентаризації, виявлення шкідливого програмного забезпечення, модуль активного реагування, хмарний моніторинг, безпеку контейнерів.

1.Модуль збирача журналів (Logcollector) призначений для збору та пересилання лог-даних від контрольованої крайньої точки вузла до менеджера системи. Модуль використовує фільтр XPath, який підходить для лог-файлів Windows, і здатний розбирати такого ж формату файли у Linux системі.

2.Команда реалізація (Command execution) – це база активних команд, що використовується агентом, яка збирає інформацію про вхідні дані та надсилає їх на сервер Wazuh. Цей модуль крім звичного моніторингу може й відобразити місткість жорсткого диску, або ж утворювати список користувачів, що прагнуть увійти в систему, які потім будуть надсилатися адміністратору, який визначить чи надавати йому доступ до ресурсу.

3.Моніторинг на цілісність файлів відомої як FIM – це процес, який займається моніторингом і виявленням будь яких змін в файлах та в файловій системі. Він регулярно контролює цілісність файлів, перевіряючи їх стан із надійною базовою конфігурацією. Загалом FIM, також відповідає за критично важливі файли, що дає змогу розсекретити будь які модифікації та підтримувати відповідність стандартам та нормам безпеки.

4.Оцінка безпеки конфігурації (SCA) запроваджується для безперервної оцінки конфігурації безпеки системного та програмного вмісту. Атрибути конфігурації перевіряються завдяки організації CIS (Центр безпеки в Інтернеті). Крім загально визначених перевірок користувачі можуть генерувати власні моделі SCA, для формування своє політики безпеки.

5.Агент інвентаризації – це компонент, що допомагає організаціям активно підтримувати актуальну інвентаризацію свої активів. Даний механізм проводить сканування інвентаризаційних даних на кшталт мережевих інтерфейсів, версії операційної системи, активних процесів, інстальованих програм та відкритих портів. Ці результати надсилаються на базу SQLite, і там же вони зберігаються. При цьому доступ до сховища можна отримати у віддаленому режимі.

6.Виявлення шкідливого програмного забезпечення (Malware detection) виконує комплексну перевірку системи для виявлення руткітів, зловмисного

програмного забезпечення, неавторизованих модифікацій, а також виявлень прихованих файлів, портів, та процесів.

7.Модуль активно реагування відповідає за автоматичне моделювання дій при виявлення загроз, таких як заблокування мережевого трафіку, обмеження процесу, усунення шкідливого файлу. Крім цього користувачам надається право на генерування власних реакцій, а також налаштовувати на свій лад. Наприклад тестування бінарного файлу в ізольованому середовищі, сканування певного типу файлів, та фільтрування мережевого трафіку.

8. Організація моніторингу безпеки контейнерів (CSM) – це модуль який працює на базі API докера, що дозволяє відстежувати будь які зміни контейнерів, виявляючи модифікації в образах, мережевих конфігурацій, чи в інших моделей. А також він видає контейнери та користувачів, які працюють від імені користувача та виконують адміністративні команди.

9. Модуль моніторингу хмари – цей елемент призначений для моніторингу машин, які розгорнуті в хмарному середовищі на сумісних з машиною платформах. З поміж категорій платформ, що налаштовуються під тип машинного продукту та підтримуються цим модулем можна розглянути такі: Amazon (AWS), Google (GCP), Microsoft Azure. В усіх платформах є API, що слугує взаємодіючим елементом так званим з'єднувачем між платформою та хмарною частиною. Цей модуль відповідає за забезпечення безпеки в хмарі, тому очевидним є те що він вирішує такі завдання як: визначають чи можна новому користувачеві надавати доступу до мережевих ресурсів, чи користувач є адміністратором, чи може він запускати команди від імені адміністратора, а також збирати інформацію про лог-файли хмарного середовища мережі.

Отже підсумувавши роботу агента в Wazuh системі можна ствердити, що агент є важливою функціональною компонентою. Завдяки йому збиратимуться дані про конфігурацію мережевих файлів та їхній онлайн статус в якому вони функціонують, після чого усі ці дані надсилатимуться до сервера Wazuh через канали зв'язку. Встановлення агента дає змогу, моніторити, оновлювати, налаштовувати роботу серверів в дистанційному режимі.

Як відомо комунікація між серверною частиною та агентурною відбувається через канали зв’язку, такі як TCP або UDP. Для безпечного постачання мережевого компоненту дані канали шифруються та стискаються в онлайн режимі. Щоб система працювала стабільно, агент врегульовує механізми потоку даних, усуваючи спам чи дублювання пакетів, що унеможливлює перенавантаження системи в цілому. Таким чином, агент є цінним інструментом в інфраструктурі системи, збільшуючи її пропускну здатність.

Перш ніж запустити роботу Wazuh платформи необхідно зареєструвати агент. Коли реєстрація агента відбулася успішно йому буде виділено унікальний ключ, який призначений для ідентифікації та аутентифікації його проміжних елементів. Крім того елементарна складова агента має бути конфіденційною, а отже, підлягати шифруванню.

2.2 Установка системи Wazuh

Для того щоб налаштувати середовище Wazuh, призначеного для пошуку вразливостей в програмному забезпеченні рекомендується встановити важливі компоненти такі як менеджер та агенти, вони слугують фундаментом на якому відбуваються усі виконавчі процеси.

Першим кроком, який необхідно зробити є встановлення менеджера Wazuh. По суті менеджер є програмним забезпечення на якому розгортатимуться агенти, які можуть застосовуватись для вирішення актуальних питань.

Менеджер Wazuh можна імпортувати у будь яке зручне середовище для користувача. В нашому випадку для програмної реалізації системи Wazuh було взято операційну систему Linux, де встановлення відбуватиметься завдяки терміналу користувача по командам взятих із документації установки. Також виділяють і альтернативні шляхи встановлення програмного продукту прикладом якого є віртуальна машина. Для цього заходять на офіційний сайті завантажують загрузочний файл із розширенням OVA, який потім імпортують у

віртуальне середовище та запускають в нім. На рисунку 2.9 проілюстровану оболонку операційної системи Kali із наявним вмістом функцій.

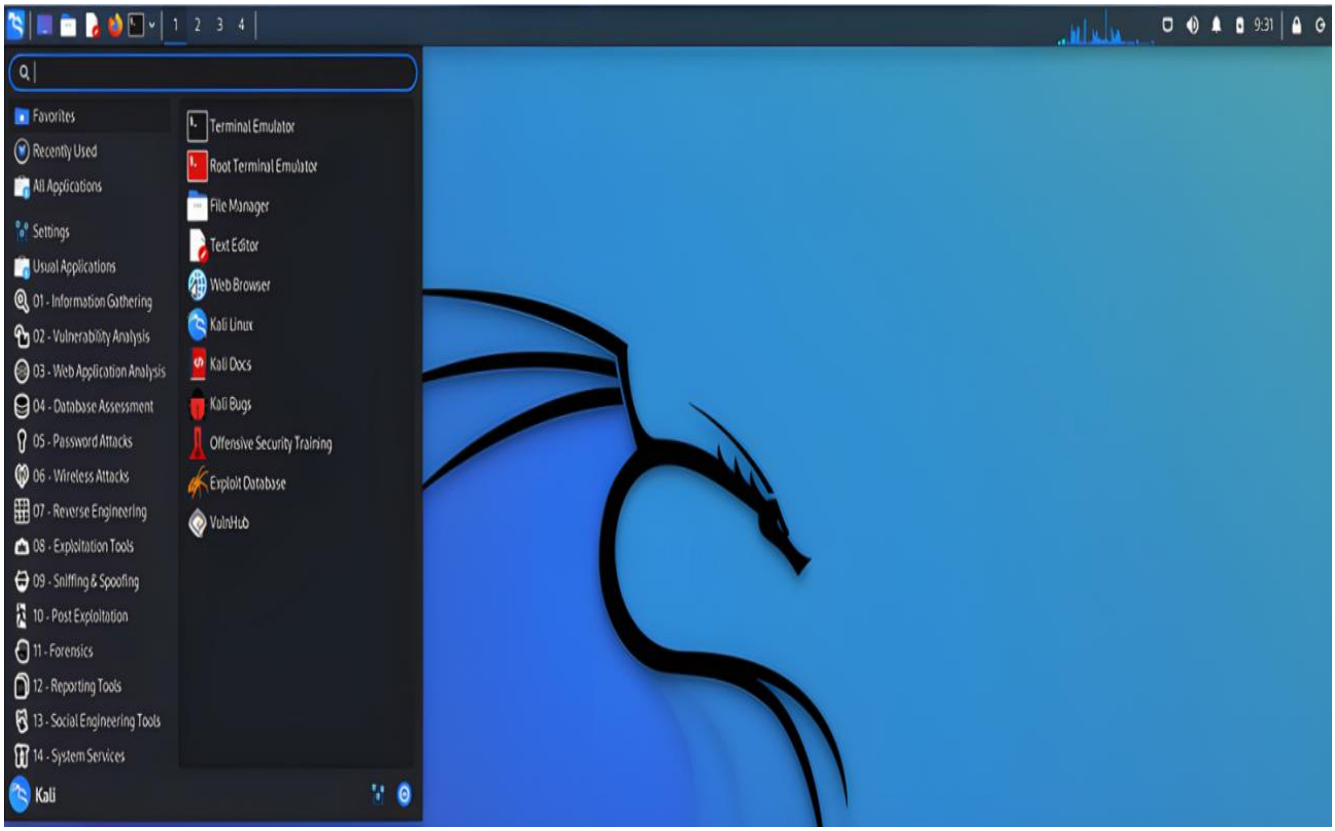


Рисунок 2.9 – Інтерфейс користувача Kali

Тому для програмної реалізації серверної частини Wazuh було вибрано віртуальну машину VirtualBox, яка славиться своїм багатим функціоналом, високою продуктивністю, адаптивністю до різного роду хостів, та постійним удосконаленням свого функціоналу до сучасних вимог.

Налаштувавши віртуальну машину на своєму персональному комп’ютері. Відкриваємо вкладку файл і натискаємо на імпортування машини (рисунок 2.10) її ми нажимаємо на іконку створити машину.

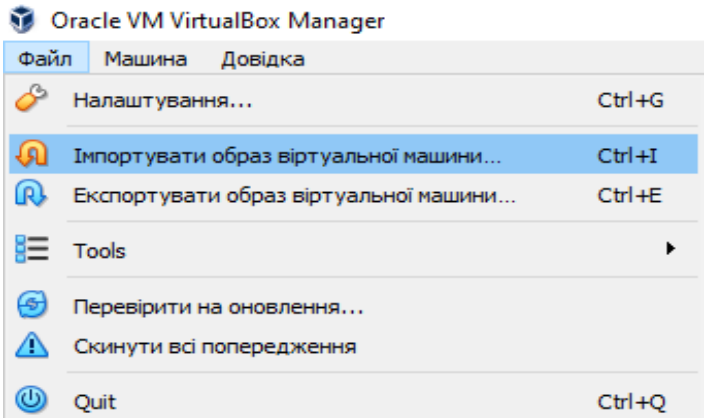


Рисунок 2.10 – Імпортування образу віртуальної машини

Даємо їй ім’я та налаштовуємо її технічні характеристики. На створеній машині будуть зберігатись усі наші файли загрузки із іншими технічними налаштуваннями. Після чого заздалегідь завантажений файл ми імпортуємо туди і запускаємо її. Далі запускається вікно, в якому даємо посилання на файл де знаходиться наша віртуальна машина Wazuh.ova.

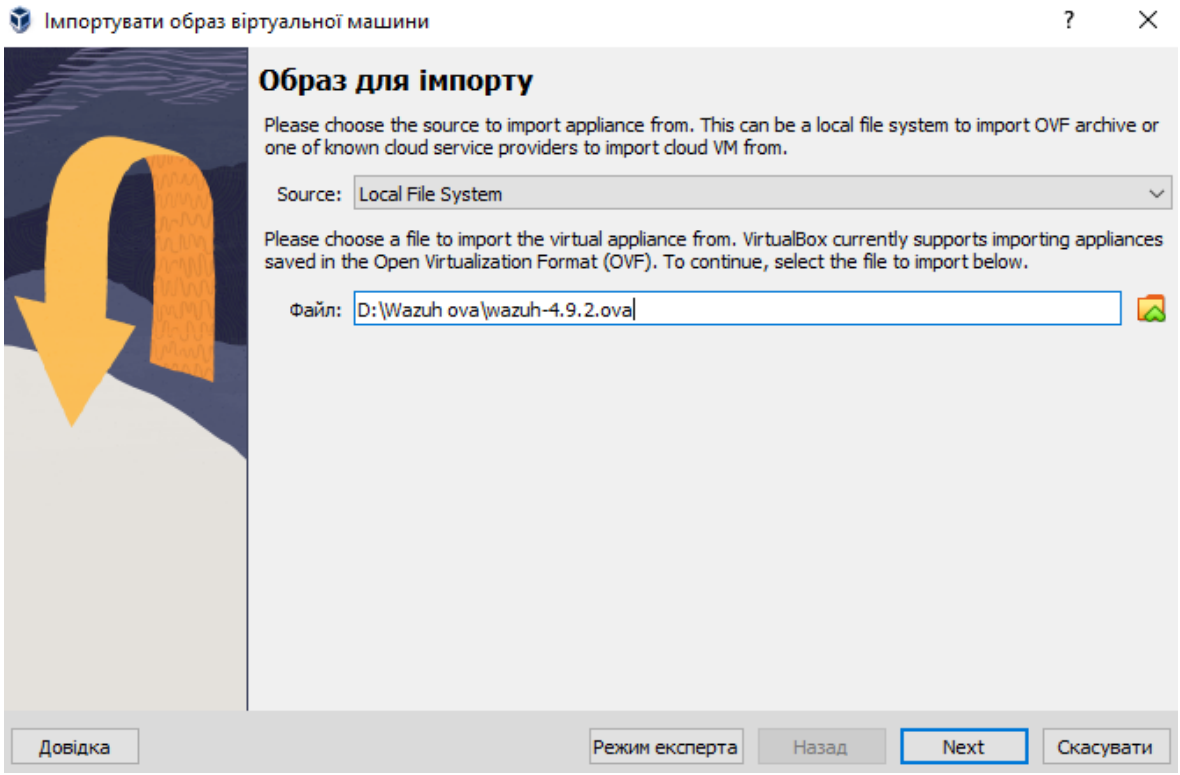


Рисунок 2.11 – Введення посилання на файл Wazuh.ova

Як тільки ми правильно ввели дані нам слід подивитись на IP адрес машини. Для цього в командній строчці ми вписуємо відповідну команду ip a. Далі ми побачимо результат виводу команди, а саме нас цікавить наша IP адреса, який має вигляд 192.168.0.102. Цей адрес свого роду декламує себе як ідентифікатор, завдяки йому користувач зможе виконати багато дій до прикладу: встановити зв'язок з агентами, надсилати звіт про стан мережі, збирати дані, а також ввівши його в браузері ми отримаємо доступ до інтерфейсу Wazuh. Таким чином на рисунку 2.12 наведено IP адрес нашої машини.

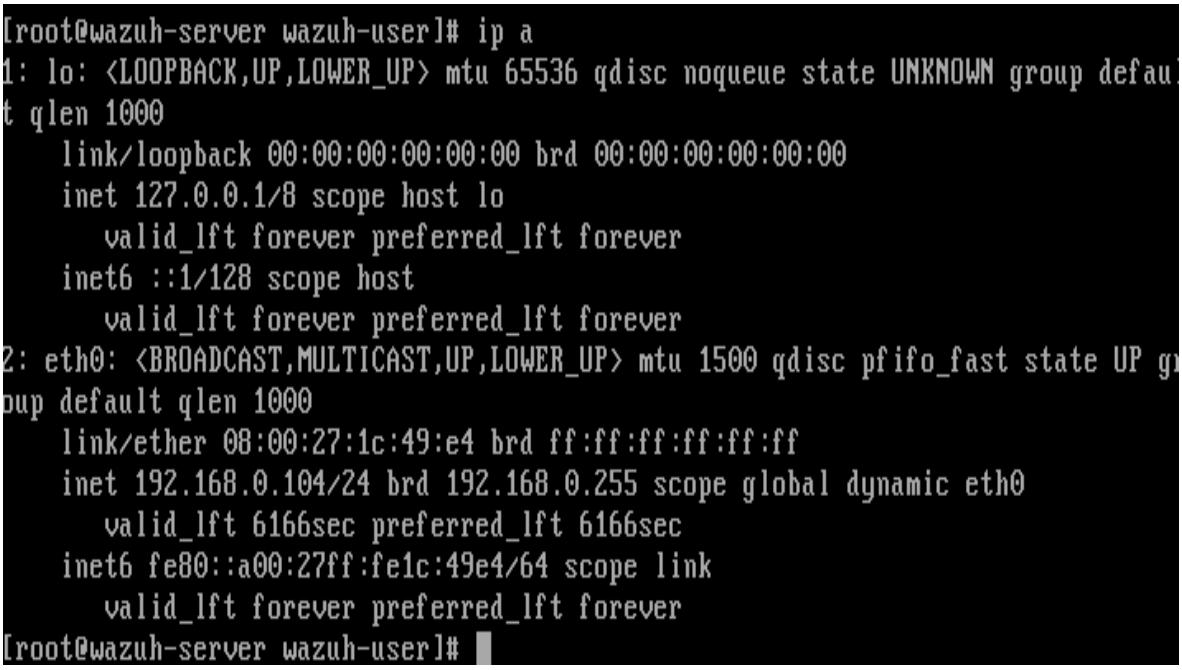


Рисунок 2.12 - IP адрес Wazuh машини

Зображену IP адресу ми вводимо в поле пошуку будь якого веб-браузера, яким ви зазвичай користуєтесь. Але не все так однозначно, для того щоб все вийшло вдало необхідно знати про один важливий нюанс, а саме в пошуковій стрічці вибраного браузера при вводі адресу слід використати посилання типу HTTPS, а не HTTP, через те, що сервер Wazuh працюватиме тільки завдяки цьому протоколу. Тому, якщо користувач намагатиметься зайти по старій версії Інтернет протоколу, йому виб'є помилка, і він не зможе належним чином зайти на сервер Wazuh.

Перед запуском сервера варто перевірити статус всіх компонентів(Wazuh indexer, manager, dashboard), і визначити в якому статусі вони перебувають. Якщо в активному режимі тоді ми зможемо все вдало запустити, але якщо в пасивному то нам потрібно їх власноруч налаштувати.

Щоб це зробити необхідно спочатку перевірити їхній статус ввівши в командному терміналі Kali команду «systemctl<wazuh-indexer, wazuh-dashboard, wazuh-manager>. Якщо з'ясується що не активний Wazuh indexer, тоді ми перевіряємо чи достатньо оперативної пам'яті ми приділили. Важливо щоб на Wazuh машину припадало половину пам'яті від загагально обсягу операційної системи Kali. Для цього ми заходимо в середену конфігураційного файлу використовуючи команду «sudo nano /etc/wazuh-indexer/jvm.options» і налаштовуємо пам'ять Wazuh машини на вигідний діапазон. Далі перезапусакаємо indexer за допомогою команди «sudo restart systemctl wazuh-indexer».

Для перевірки статусу навігаційної панелі скористаємося командою «sudo systemctl wazuh-dasboard». Тоді якщо виявиться що панель не активна, заходимо в файл установки завдяки цій функції «sudo nano /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml» і редагуємо його вміст за вказаними підказми що є в ньому і перезавантажуємо dasboard «sudo restart systemctl wazuh-dashboard».

Ще одним із важливих комнентів, статус якого необхідно перевірити це Wazuh maneger. Для цього коритустувачеві слідпрописати в терміналі таку команду«sudo systemctl wazuh-manager». Вразі не активно статус manager потрібно лише перезапусти застосувавши таку команду «sudo restart systemctl wazuh-manager».

Виконавши всі ці процедури ми остаточно переконайомося що все налаштувалося правильно, адже якщо навіть один компонент Wazuh знаходитиметься в пасивному статусі, той все інше не буде працювати, тому користувач не зможемо увійти в середину цієї машини і зробити будь який

дослід. На рисунку 2.13 наглядним чином показано статуси головних компонентів Wazuh.

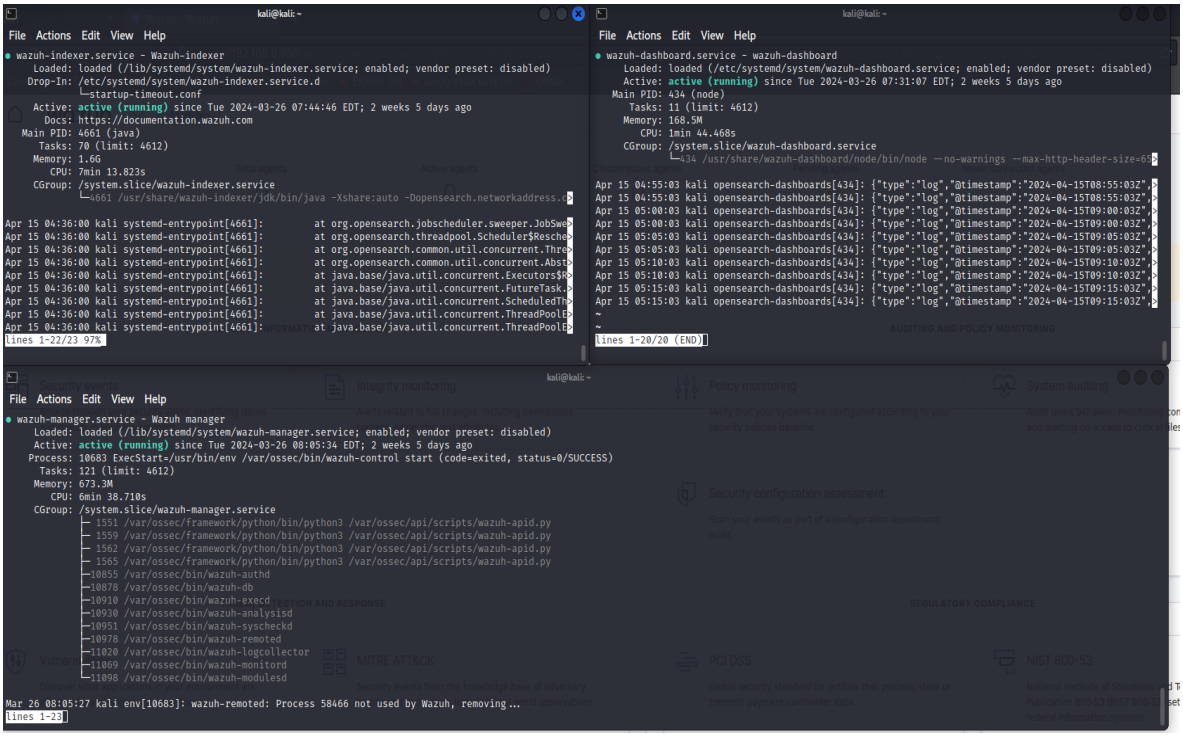


Рисунок 2.13 – Перевірка статусів головних серверних компонентів

В разі успішного запуску усіх головних процесів середовище Wazuh працюватиме в звичному режимі. Для початку необхідно ввести IP, його ми можемо взяти в результаті установки файлу Wazuh із розширенням ova, або ж присвоїти йому адрес операційної системи Linux. Завдяки введеній URL адресі користувач має пройти аутентифікацію. Конфіденційні дані йому надаються в результаті завантаження установчих файлів, тому потрібно бути уважним, адже вони є важливими при вході в адмін-панель.

Якщо всі дані введено вірно ми заходимо в інтерфейс користувача (рисунок 2.14) в якому є різноманітний функціонал, який ефективно виявляє та запобігає шкідливому програмну забезпечені. Користувальницька панель є доволі простою та зручною це ми бачимо відразу ж при вході в дане середовище. де користувачеві пропонується встановити агент. В ній все гарно добре розписано і розподілено за виконавчими параметрами. В ході роботи користувач зможе більш детальніше ознайомитися із запропонованим функціоналом. А так

практично дана система наділена гнучким інтерфейсом, який дає змогу користувачам налаштовувати Wazuh середовище під власні потреби, така властивість робить його зручним продуктом у застосуванні.

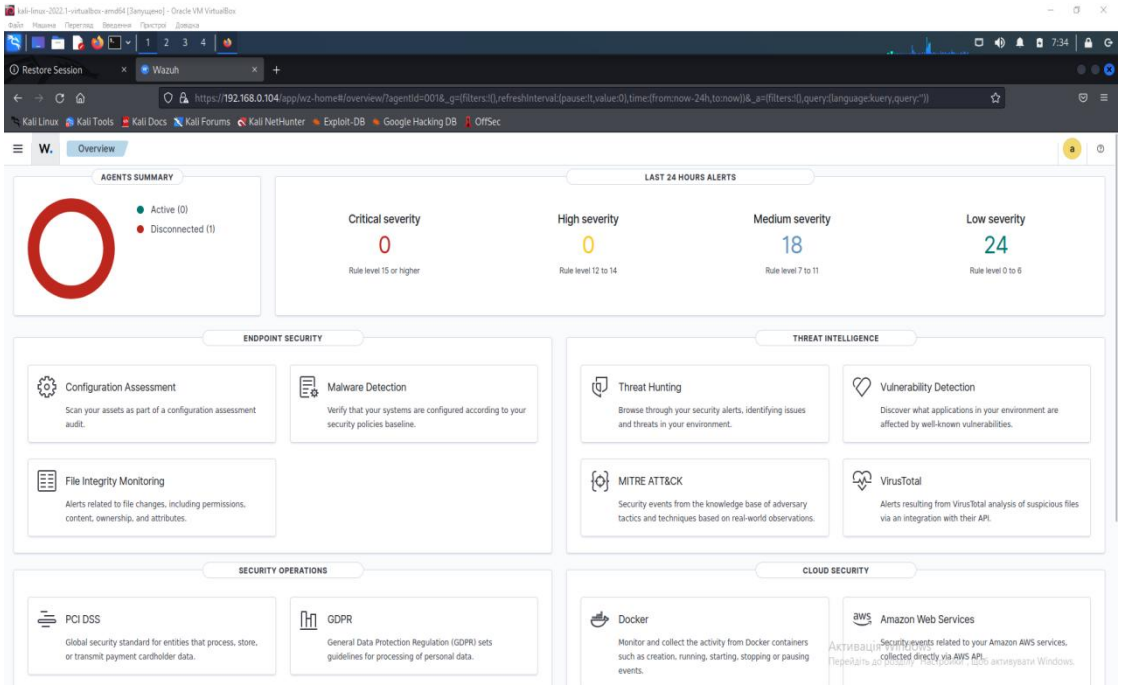


Рисунок 2.14–Інтерфейс користувача Wazuh

2.3 Моделювання системи зловмисника

Вибравши операційну систему потенційної жертви необхідно на противагу їй обрати систему зловмисника для того, щоб ми змогли належним чином протестувати систему Wazuh та побачити на практиці методи її боротьби із шкідливим програмним забезпеченням. В ролі системи зловмисника буде ж та сама операційна системи тільки іншої версії.

Установка та налаштування даної системи є доволі простою. Тому користувач може з легкістю встановити даний продукт чи на свій пристрій чи на віртуальну машину чи будь яким іншим методом. На офіційному сайті розробника ми може знайти багато інформації про дану платформу та дослідити її особливості використання, а структурований виклад інформації в документації робить цей ресурс зрозумілим для користувачів.

Kali Linux – це платформа, специфіка якої заключається в тестуванні на проникнення операційних систем та запровадженні нових безпекових заходів для її захисту. Ця платформа розроблена на основі Debain дистрибутива із широким набором інструментів та програм. Тому операційна систем Kali є затребуваною не тільки серед кіберфахівців а й студентів, які використовують її для здобуття досвіду, що дає змогу їм оцінити безпеку IT систем шляхом аудиту, тестування тощо.

В порівняні із іншими дистрибутивами Kali наділений такими особливостями: гнучким та простим налаштуванням, насиченим інтерфейсом, безпекою та пентестингом, технічною сумісністю зі іншими обладнаннями, та частим оновленням.

Якщо в користувача виникнуть якісь питання він звертається до розробників даної системи або ж написати в ком'юніті спільноти користувачів. Відповідно до норм безпеки вхід в платформу відбувається лише за дозволом користувача, в якому він встановлює свої рамки безпеки, визначаючи кому надавати доступ до системи або кого обмежити.

Для установки Kali потрібно завантажити образ віртуальної машини, серед доступних варіантів – Virtual Box, VMware, QEMU та HYPER-V. У цьому випадку буде використано Virtual Box. Потім потрібно завантажити і розпакувати архів, в результаті чого з'явиться жорсткий диск VDI, що містить Kali Linux.

Наступним кроком буде запуск VirtualBox, який постачається з попередньо встановленими компонентами VBox Guest Addition sand Extensions Pack [13]. Потім натискаємо на меню "Створити", вибираємо тип ОС (у нашому випадку Linux, Debian), встановлюємо обсяг оперативної пам'яті і вибираємо жорсткий диск VDI для імпорту.

Щойно додана віртуальна машина з'явиться на головній інформаційній панелі, де можна налаштувати параметри і ввімкнути або вимкнути необхідні опції. Це може включати вимкнення звуку, налаштування мережі (в тому числі налаштування NAT), підключення спільної папки для спільного доступу до

файлів тощо. Після встановлення всіх необхідних конфігурацій віртуальну машину можна запустити.

Візуалізація системи є досить варіативною із різноманітними наборами робочих областей серед яких розглядають такі оболонки як Xfce, Gnome, KDE, кожен з них має свої особливості та властивості [14].

Xfce – класичне середовища UNIX систем, завданням якого є зробити компактне та зрозуміле середовища для користувача [14]. Дане середовище легко встановлюється на старе обладнання або не системи із обмеженими ресурсами. Робочу область середовища Xfce легко настроїти та адаптувати під власні уподобаннякожна з яких виконує притаманні іоперації в конкретнійробочій області.

GNOME – оболонка Kali, що відзначається зручним та сучасним інтерфейс користувача акцент якого орієнтований на простоту та легкість його використання [14]. Практично користувач зможе з легкістю перемикатись між робочими областями та утилітами. На відміну від робочого стола XFCE оболонка Gnome затребує більш системних ресурсів, а тому вона підійделише для потужного обладнання.

KDE – плазма, яка характеризується широким набором технічних можливостей [14]. Гнучкий функціонал надаєкористувачам можливість налаштовувати інтерфейс під власний стиль, змінюючи робочий фон, шрифт, та розташування віджетів на робочому столітощо.Візуально дане середовище наділене розширеними ефектами та анімацією. Порівнюючи XFCE та GnomeShell можна сказати що, дана оболонка є більш ресурсномісткою, тому вона вимагає більш технічно оснащеного обладнання.

В підсумку XFCE є ефективними та легким середовищем у використанні, Gnome Shell візуально привабним та компактним ресурсом, тоді як KDE Plasma має широкі можливості налаштування із багатим функціоналом. Зрештою вибір робочого столу залежить від персональних потреб та вподобань користувача.

Ця операційна система наразі перебуває на стадії постійного розвитку, конкуруючи із провідними системами такими як Microsoft Windows та Apple,

розширюючи свої функції та інструменти в майбутньому, одночасно вдосконалюючи свої існуючі можливості. Проте зіставляючи із іншими операційними системами неможливо остаточно виділити ту систему яка є кращою з усіх по всім параметрам, адже у кожної з них є як переваги та і недоліки. Тому в таблиці 2.1 наведено порівняльний аналіз даних систем за їхніми особливостями [7].

Таблиця 2.1 – Порівняльний аналіз даних систем за їхніми особливостями

ОС	Ліцензія	Налаштування та конфігурація	Використання на практиці	Програмні засоби	Рівень безпеки
Linux	Безкоштовна	Потрібно бути обізнаним у взаємодії між програмними пакетами	Не простий у використані	Поширені та у відкритому доступі	Високий
MacOs	Безкоштовна для Apple продукції	Інтуїтивно легкий інтерфейс користувача	Максимально зрозумілий у використані	Не велика кількість програм у вільному доступі	Високий
Windows	Платна	Проста у налаштуванні	Зручний та комфортний інтерфейс у використані	Багато корисного софту в мережі Інтернет на платній основі	Низький

З таблиці 2.1 ми можемо сказати що всі операційні системи широко використовуються у наш час. Проте все ж таки мають і свої особливості. Так, якщо аналізувати ліцензійний аспект, то переваги мають Linux та MacOs над Windowsадже вона платна і вимагає постійної активації. Однак налаштування систем Windowsта Маснабагато простіше ніж у Linux.

Якщо порівнювати інтерфейси користувачів, то можна сказати що у Linuxбільш складніший інтерфейс, тому потрібний час, щоб його освоїти. За поширенням системних програм у вільному доступі лідером є Linux. Якщо оцінювати рівень безпеки кожної системи, то можна дійти до висновку, що Linuxта MacOs є більш захищеними, на відміну від Windows, це тому що в Linux

права адміністратора отримують обмежене коло осіб, а у Windows отримують за замовчуванням, тим самим будь який користувач може розпоряджатись всією системою. Також вагомий вплив має попит операційної системи серед користувачів, як відомо систему Windows використовують більше користувачів аніж Linux. Linux має розгалужену систему дистрибутивів та велику аудиторію спеціалістів, які оперативно виправляють помилки в коді.

3. ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ЗАГРОЗ МАШИНОЮ WAZUH

У цьому розділі сфокусована увага на практичному застосуванні Wazuh. На практиці ми ознайомимось з поширеними методами боротьби з шкідливим програмним забезпеченням, ефективно виявляючи інциденти та завчасно реагуючи на їх вторгнення. В якості шкідливого продукту виступатиме Ransomware. Тож завдяки даному ПЗ буде сформульовано детальний опис протидії Ransomware. Також буде приділено увагу процесу оповіщення про ймовірні загрози, та налаштовано те середовище на яке вони надходитимуть. Для проведення досліду буде розроблено моделі Ransomware атак та застосовано інструменти Wazuh, що займатимуться тестуванням системи на проникнення через сканування та запровадженню атак завдяки віртуальний машині зловмисника на цільову жертву системи.

3.1 Розгортання агента Wazuh

Коли нам вдалося успішно розгорнути інтерфейс користувача, то подальшим кроком, який варто зробити це зайти на офіційний сайт розробника на якому розміщено послідовну інструкцію установки моделей агента під певний тип операційної системи і звичайно слідувати їй. В нашому випадку програмне забезпечення агента буде установлено на операційну систему Kali Linux.

Насамперед сам агент можна встановити двома способами, а саме набором команд в терміналі, які є в документації, або ж пряме заповнення даних про агент та пристрій на якому він встановлений. Розглянемо більш детальніше обидва методи.

Отже якщо ми налаштували його через консоль, то як встановиться агент нам відкривається його інтерфейс. Далі ми мусимо нашому агенту присвоюємо IP адрес менеджера Wazuh (та сама адреса що використовувалася для запуску програмного забезпечення Wazuh), і вказуємо йому вхідний ключ.

Якщо ж ми заповнюємо форму яка безпосередньо розміщена за посиланням «ADD agent» або в тогочасній версії потрібно зайти управління сервером (Server Management) і вибрати Endpoints summary, таким чином нам вдасться відкрити форму агента.

Коли користувач встановлює агент шляхом заповнення форми ми вказуємо на архітектуру операційної системи. В даному випадку агент установлюється на Linux DEB amd64 архітектури. Для комунікації агента із машиною Wazuh вписується його IP адрес, що показано на рисунку 3.1.

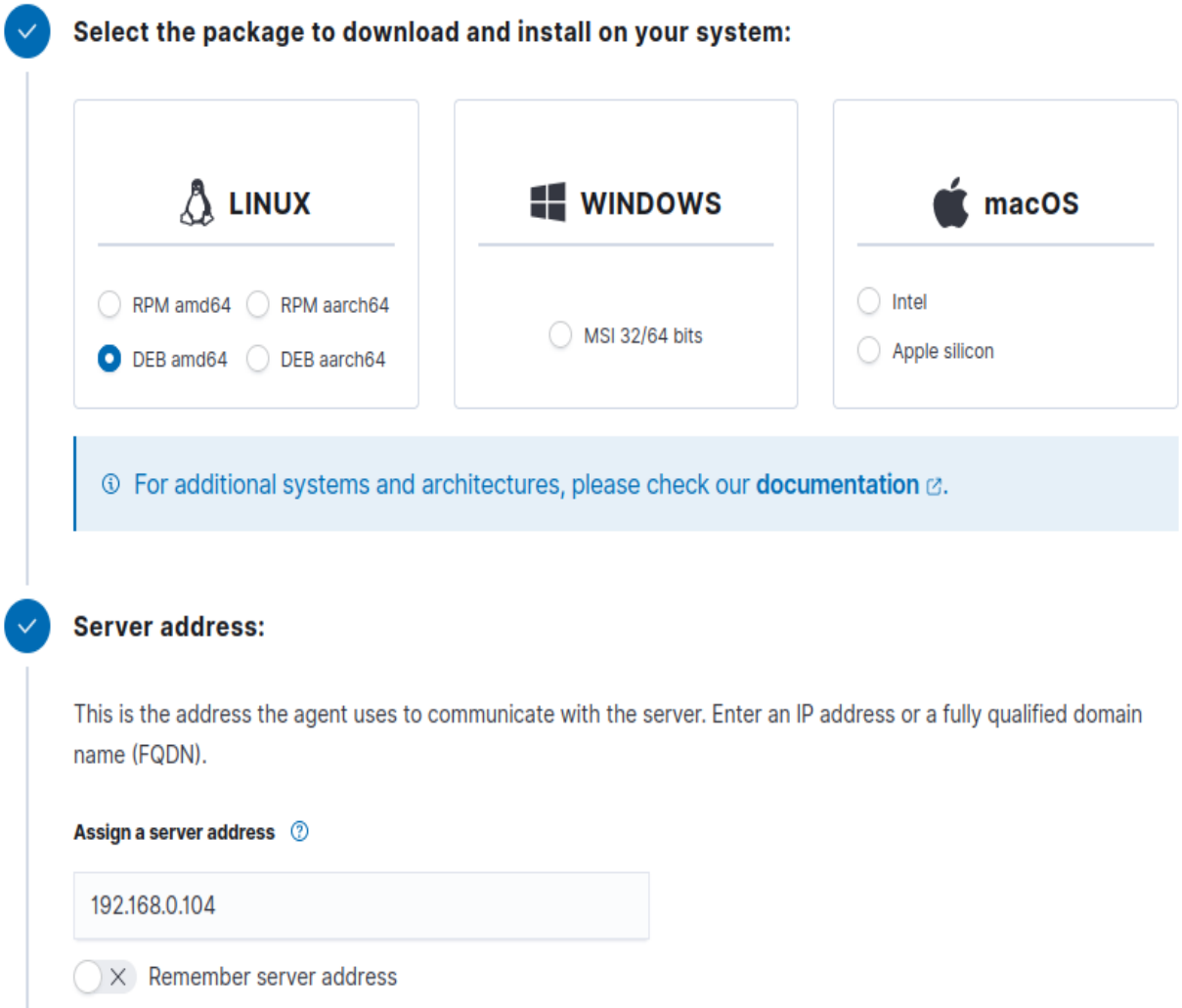


Рисунок 3.1 – Визначення архітектури операційної системи та налаштування серверного адресу між агентом та оболонкою Wazuh

Далі ми присвоюємо агенту унікальне ім’я “Kali_agent_Nazar”, для того щоб в подальшому не виникало плутанини між агента, і ми б з легкістю могли

б ідентифікувати тип агента та його приналежність до певної операційної системи.

В комірці груп залишаємо значення “default”. Та бачимо сформовану команду із введеними даними, що зазначалися раніше. На рисунку 3.2 зображено додаткові налаштування та команду, яка інсталює агента.

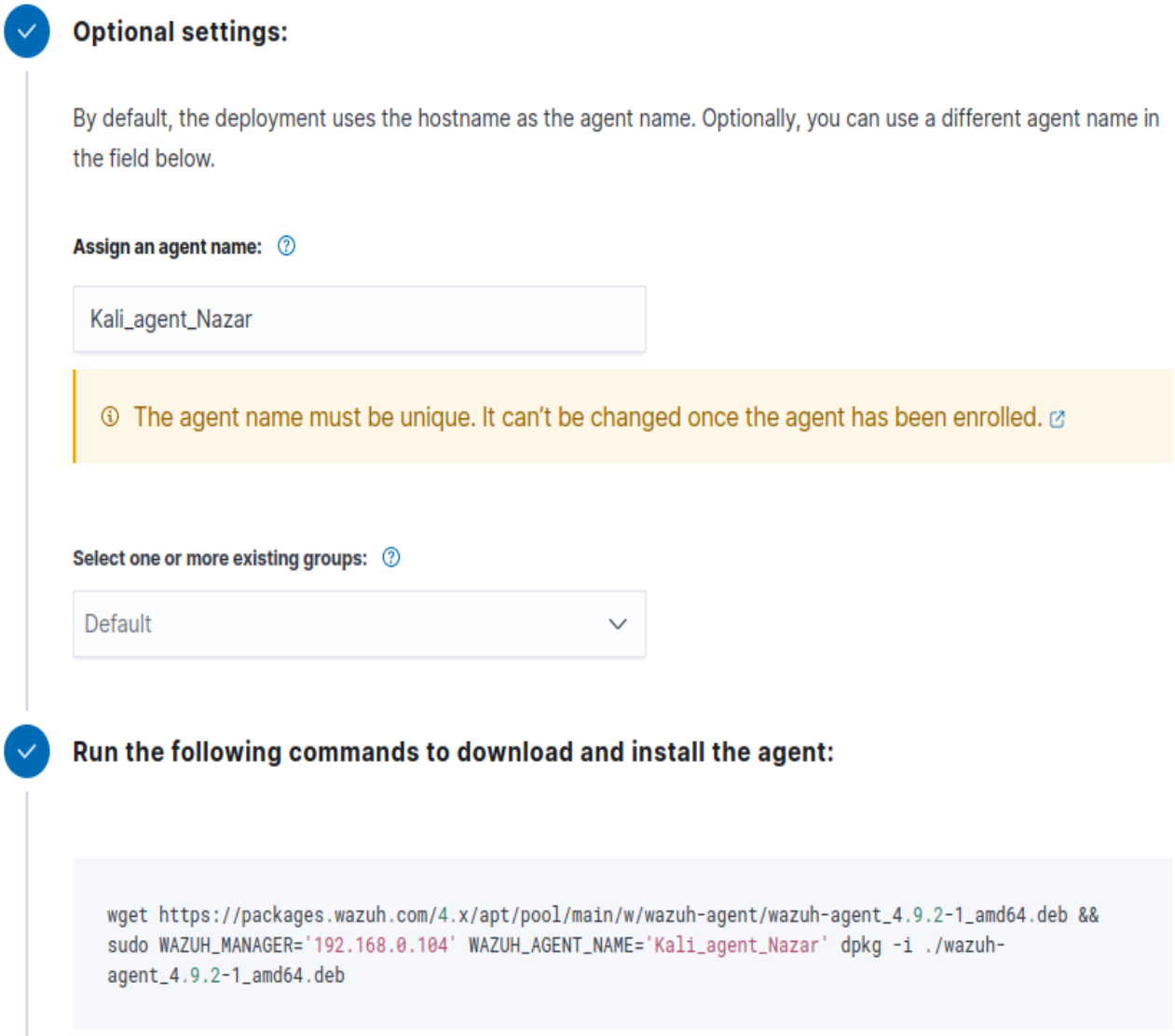


Рисунок 3.2 – Додаткові налаштування та інсталяція агента

Після завантаження агента, нам потрібно його запустити, вводячи наступні команди, які наведено на рисунку 3.3.

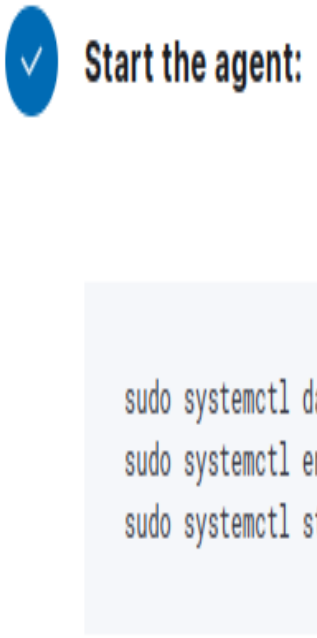


Рисунок 3.3 – Запуск агента Wazuh

Варто зауважити, щоб дана команда спрацювала необхідно запустити іншу операційну систему в нашому випадку було вибрано систему Kali Linux версії 2021 та було скопійовано в термінал дану команду, і вона успішно виконалась. З іншого боку, якщо встановити агента в тій же операційній системі, в якій перебуває платформа Wazuh, ми не зможемо активувати його. Це пов'язано з тим що для проведення аналізу потрібно використовувати дві операційні системи, при цьому кожна з них займає свою позицію. Перша виступатиме з позиції спостерігача а друга з позиції досліджуваного об'єкту. На рисунку 3.4 наведено процес завантаження пакетів агента та його ввімкнення в операційне середовище.

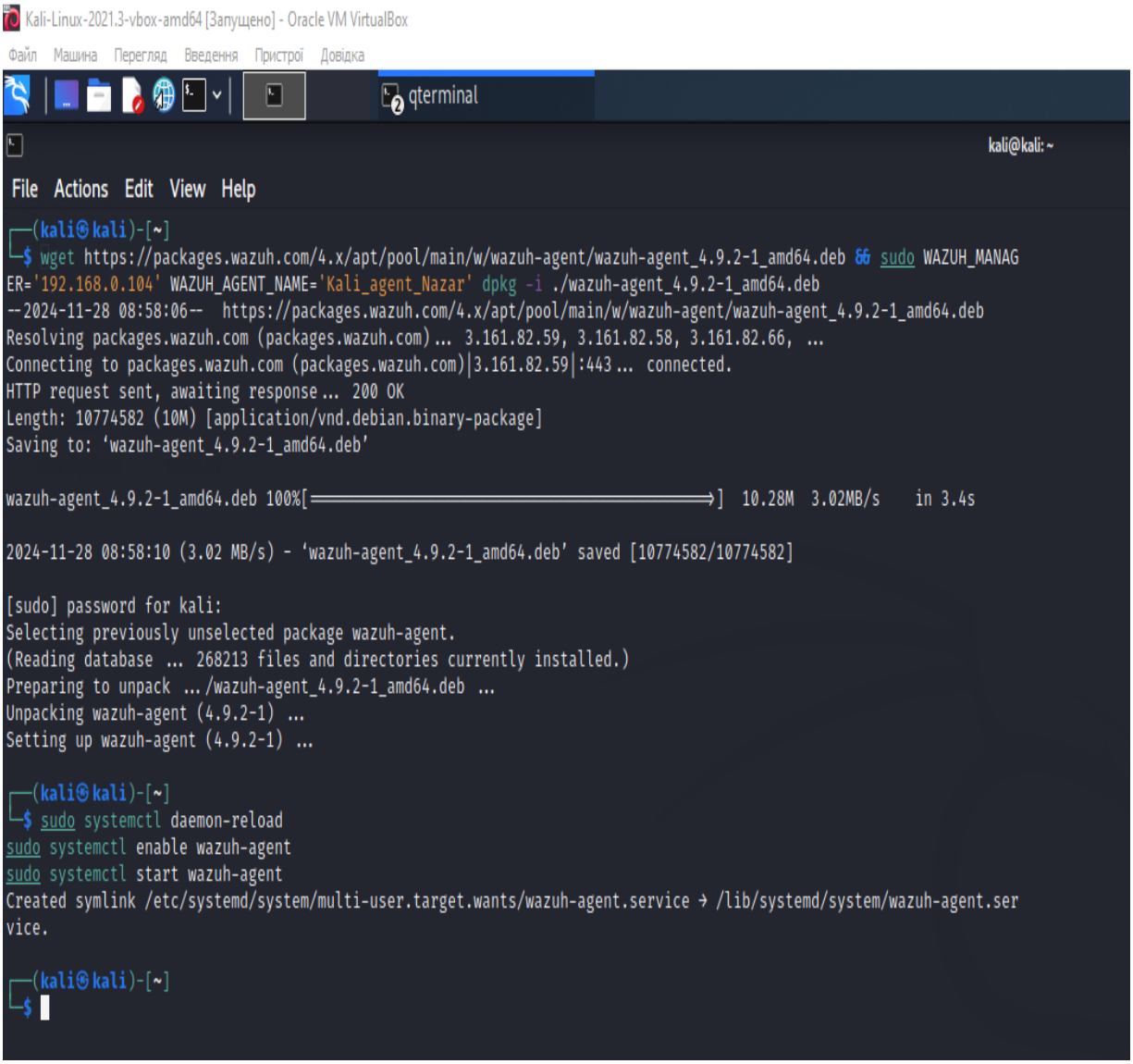


Рисунок 3.4 – Інсталяція та запуск агента в іншій операційній системі

Як ми бачимо на малюнку показано конфігурацію агента з його важливими даними. Після того як ми заповнили форму на панелі користувача з’явиться інформація про статус нашого агента (рисунок 3.5).

В цю панель можливо підключити велику кількість агентів, а отже користувач зможе активно прив’язуватись до операційних систем та досліджувати їхню поведінку. Від агентів користувач отримує повну звітність активних подій, які виконувались системою вході яких користувач формуватиме висновки щодо посилення безпеки.

В цілому Wazuh агент є багатофункціональним продуктом, який може виявляти шкідливі файли в структурі лог-файлів, перевіряти активний трафік на

відповідність нормам політики безпеки, та постійно моніторити систему в цілому.

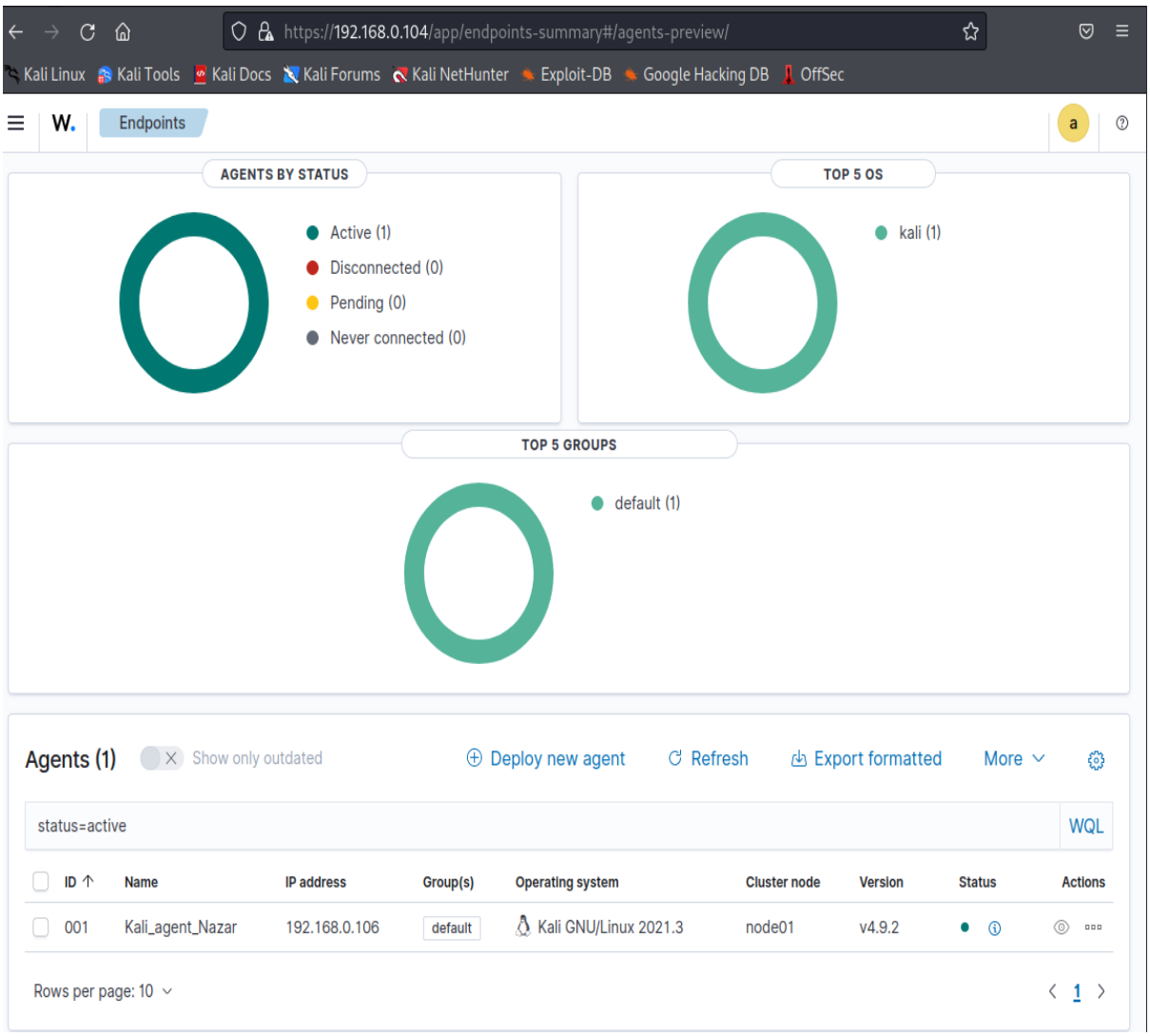


Рисунок 3.5 – Перелік активних агентів

Встановивши агент та перевіривши його на справність ми приступаємо до обрання цільової системи на яку будуть скомпрометовані атаки. Таким чином в ролі жертви вибрано операційну систему Kali попередньої версії, яка об’єктивно схожа до цільової системи зловмисника, основна відмінність між ними за версіями, однак ці системи знаходитимуться в спільній підмережі. На рисунку 3.6 показано вміст “Kali_agent_Nazar” агента.

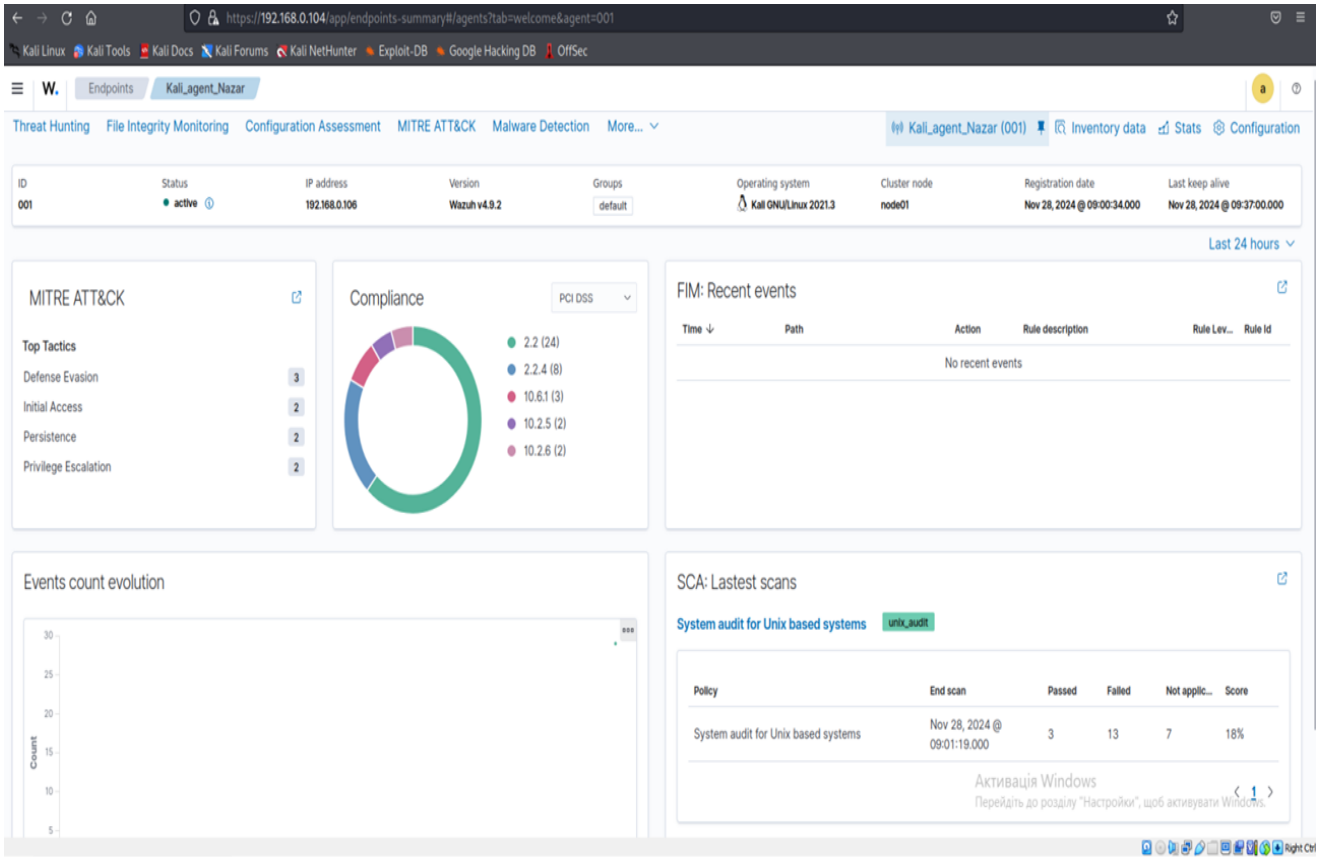


Рисунок 3.6 – Внутрішній вміст агента “Kali_agent_Nazar”

3.2Моніторинг файлової системи за допомогою Wazuh

Для забезпечення комфортного та безпечного робочого середовища користувачеві необхідно завжди підтримувати кібергігієну в усіх аспектах системи. У цьому підрозділі буде піднято питання щодо виявлення та запобіганню потенційних атак програм-вимачів націлених на модулі файлової системи шляхом їхнього моніторингу.

Даний тип шкідливого забезпечення працює за принципом шифрування відповідних комп’ютерних програм чи даних, відновлення яких відбувається після сплаченню викупу.

Станом на 2024 рік поширення програми вимагача збільшилося. Це пов’язано з тим, що вони стали використовувати різноманітні шляхи зараження. Серед яких виділяють соціальну інженерію, системи що не оновлюються, обхід

багатофакторної аутентифікації, компроміси ланцюга поставок, уражені USB-накопичувачі [8]. Крім цього вони здатні використовувати експлойти що дозволяє поширити шкідливий вміст між іншими системами у мережі. Для того щоб убезпечитись від програм-вимачів рекомендується часто оновлювати програмне забезпечення, запроваджувати додаткові рівні захисту, робити часті аудити, створювати резервне копіювання, та регулярно навчати персонал кібербезпеці.

Для ефективного виявлення та запобігання програмам-вимачам застосовують такі модулі:

- виявлення вразливості системи без сканування: ідентифікує шкідливе програмне забезпечення співвідносячи дані із CVE;
- перевірка норм безпеки: переглядають кожний компонент системи та звіряють із зазначеними стандартами CIS (Цент Інтернет безпеки);
- моніторинг файлової системи: веде спостереження за файлами та виявляє будь які шкідливі файли в системі.

Розглянемо на практиці дії які виконує програма-вимач під час атаки на файл:

- копіює вміст файлу;
- вміст файлу шифрує та перенаправляє його в новий зашифрований файл;
- видаляє початковий файл.

Розгортання Wazuh на цільову основну машину користувача дозволяє відстежувати події файлової системи, надаючи користувачеві контроль над поведінкою файлів у його операційній системі. Цей контроль дозволяє виявити програми-вимагачі, які можуть бути ідентифіковані за надмірною кількістю додавань нових файлів або ж навпаки видаленням вихідних файлів. Користувач буде повідомлений про ці дії за допомогою системи оповіщення Wazuh, що спонукатиме його вживання відповідних контрзаходів для нейтралізації цього типу програмного забезпечення.

Для того щоб виявити програми-вимагачі за допомогою платформи Wazuh для початку буде доцільним зімітувати сценарій Ransomware створивши шкідливий програму та встановивши його в файлове середовище. Сценарій вимагає підтримуваної версій Python 3 з можливістю інтеграції усіх основних пакетів бібліотеки cryptography.

Після успішної підготовки та імпорту файлу з сценарієм Ransomware на цільову машину нам потрібно розробити тестове середовище, в якому ми створюємо каталог із шляхом /home/vagrant/test, де міститимуться наші файли, та підключаємо агент налаштувавши його конфігурацію /var/ossec/etc/ossec.conf.

Для того щоб приєднати конфігураційний файл Wazuh до конкретного каталогу ми маємо зайти в середину конфігураційного файлу та прописати в ньому функцію моніторингу who-data, яка призначена для збору, обробки та моніторингу інформації в бажаному каталозі.

Зазвичай дана функція виконує роль аудиту в системі збираючи дані файлів, що розташовані в каталозі та імпортуючи їх на сервер Wazuhмашини. Встановлення аудиту на вузли системи розроблених на основі Debian відбувається наступним чином. В командному терміналі встановлюємо демоверсію аудиту за допомогою команди “apt-getinstallauditd” та додаткові до нього плагіни “apt-getinstallaudispd-plugins”.

Після завантаження необхідної конфігурації можна встановити з'єднання та розпочати моніторинг функції Who в межах каталогу, попередньо вказавши її значення. Виконання цієї команди дозволить користувачеві в режимі реального часу отримувати структуровану зведену інформацію про стан компонентів файлової системи, яка генерується інтегрованим ядром операційної системи.

Для того, щоб впровадити вищезгадані зміни, необхідно перезапустити агент Wazuh за допомогою команди "systemctl restart wazuh-agent". Це призведе до активації правила аудиту для контрольованого каталогу. Щоб переконатися в успішному встановленні правила аудиту, слід виконати наступну команду: "auditctl -l | grepwazuh_fim". Якщо з'явиться результат, що вказує на те, що правило аудиту успішно встановлено, то процес встановлення було успішно

реалізовано. Аналогічним чином, якщо служба Wazuh не працює, правило можна видалити і перевірити його видалення таким же чином.

Наступним кроком буде виконання скрипу програми-вимагача із заздалегідь написаним кодом. Експеримент передбачатиме тестування файлів, які ми необхідно імпортувати в каталог. Щоб розпочати цей процес треба ввести в терміналі таку команду “python3 wazuh-ransomware-poc.py prepare”. Ця команда автоматично заповнить наш каталог новими директоріями та файлами, кожен з яких матиме певні розмірами та значеннями. Тоді ми переглядаємо вміст нашого каталогу використовуючи команду ls -lRh /home/vagrant/test/ і на рисунку 3.7 побачимо результат.

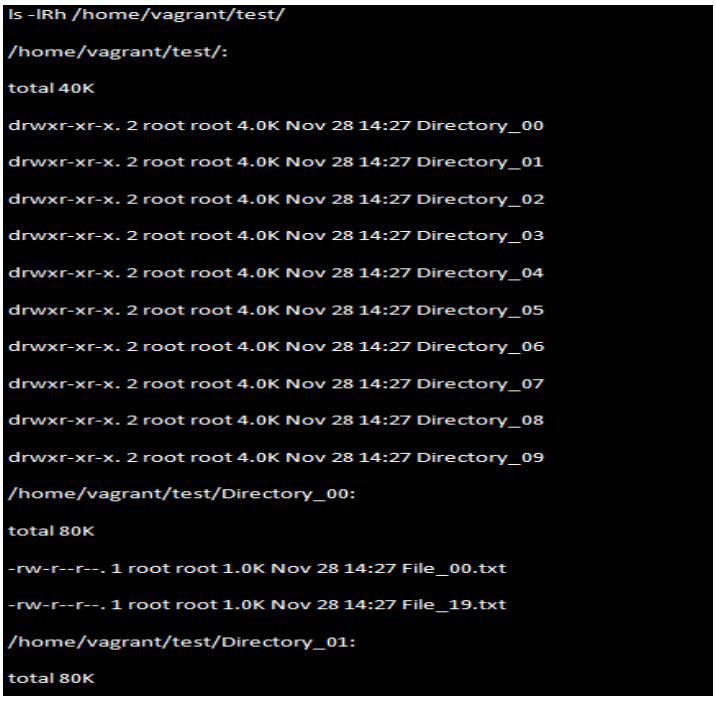


Рисунок 3.7 - Перелік імпортованих директорій та файлів

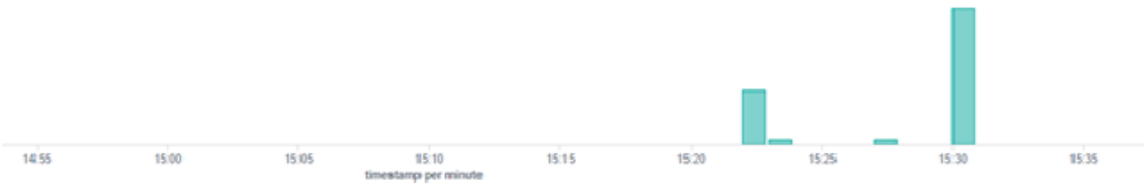
Щоб прив'язати агента Wazuh до каталогу Linux, ми повинні спочатку встановити та налаштувати агента Wazuh на основну операційну систему. Після цього необхідно виконати наведені нижче кроки, для того щоб прив'язати агента до конкретного файлу в каталозі Linux:

1. Відкрити оболонку термінал в системі Linux.
2. Перейти до відповідного каталогу, в якому встановлено агент Wazuh.

3. Виконайте команду прив'язки агента до цього каталогу, вказавши необхідні параметри, такі як ім'я сервера або IP-адреса сервера керування Wazuh.

Встановивши зв'язок між агентом і кожним елементом каталогу, користувач зможе отримати вичерпну інформацію про кожен файл у каталозі та визначити всі наявні в ньому вразливості. Дані будуть зібрані та належним чином надіслані на сервер Wazuh.

Після виконання вищезгаданих кроків платформа Wazuh зможе точно виявляти будь-які зміни у файловій системі, такі як встановлення нових файлів або видалення оригінальних файлів. В результаті, при вході в інтерфейс, користувач побачить список нещодавно виявлених файлів (Малюнок 3.8). Важливо розуміти, що кожен файл у каталозі відстежується відповідним агентом, який з ним пов'язаний.

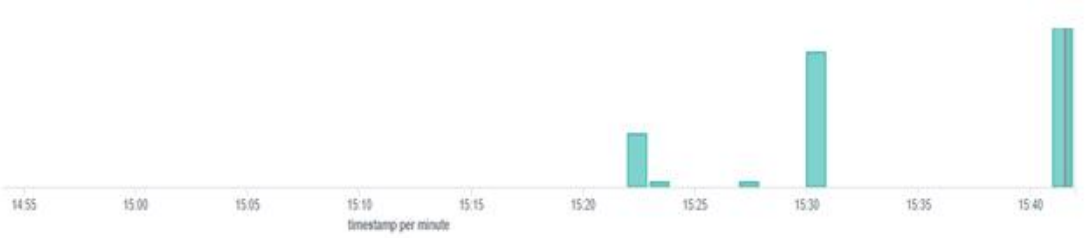


full_log	rule.description	syscheck.event
File '/home/vagrant/test/Directory_09/File_19.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_18.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_12.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_15.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_14.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_13.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_06.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_09.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_10.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_08.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_00.txt' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_03.txt' was added.	File added to the system.	added

Рисунок 3.8 - Відображення файлів каталогу на інформативній панелі

Для того, щоб проілюструвати поведінку сервера Wazuh у випадку атаки зловмисників, необхідно написати в терміналі наступну функцію: У терміналі слід ввести наступну команду: "python3 wazuh-ransomware-poc.py attack". Після

виконання цієї команди почнеться процес дублювання файлів, в результаті якого існуючі файли будуть видалені, а нові зашифровані. Згодом на інформаційній панелі Wazuh з'явиться повідомлення про виявлення подій, що відбулися під час моделювання цієї атаки. В інтерфейсі користувача показаного на рисунок 3.9 ми спостерігатимемо за моніторингом файлової системи, який відображатиме зображення перезапису файлів і бачитимемо деякі відмінності із попереднім рисунком, які виділені кольором що сигналізують про додавання “added” та видалення “deleted” оригінальних файлів.



full_log	rule.description	syscheck.event
File '/home/vagrant/test/Directory_09/File_18.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_19.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_17.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_17.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_16.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_15.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_14.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_11.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_11.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_08.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_07.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_08.txt.encrypted' was added.	File added to the system.	added

Рисунок 3.9 - Моніторинг файлової системи після атаки програм-вимагачів

3.3 Налаштування моніторингу та тригерів сповіщень Wazuh

Платформа Wazuh чудово зарекомендувала у процесі виявлення шкідливого програмного забезпечення. Вбудований модуль оповіщення здатний

ефективно повідомляти користувача про будь які атаки в тому числі і ті про які він і не знав.

В системі Wazuh є багато методик оповіщень серед яких розглядають можливість відправки повідомлення електронною поштою. Робота електронної пошти потребує використання локального SMTP сервера. Для його налаштування потрібно завантажити ще одну операційну систему наприклад Ubuntu, дистрибутив якого успішно кооперується з системою Wazuh. Для того щоб сервер працював необхідно завантажити відповідні пакети, а саме:

- Postfix – агент що виконує функції доставки сповіщень на електрону пошту та є основою сервера для відправки листів SMTP;
- Cacerficates;
- Libsasl2-2;
- Libsas2-modules;
- Mailutils.

Як ми вже зазначали ключовим елементом нашого SMTP сервера є Postfix, а всі решта перераховані компоненти являються лише додатковими утилітами, що підтримують стабільність та розширюють область дії агента. Результати роботи агента надсилатимуться на пошту електронного сервісу Gmail розробленого компанією Google, але перш за все потрібно зареєструвати акаунт та ввести свої облікові дані. Тоді під’єднуємо сервер частину SMTP до нашого аканту завдяки підключеню другого фактору автентифікації.

В конфігураційному файлі агент Postfix потрібно зробити такі дії:

- налаштувати автентифікацію на smtp.gmail.com, для того щоб відправленні листи автоматично спрямовувались до користувача;
- активувати TLS та підключити сертифікати із пакету ca-certificated;
- додати IP адресу нашої Wazuh машини до змінної мережі “my_networks” для того щоб агент Postfix зумів ідентифікувати наш менеджер. та обробляти листи надіслані платформою Wazuh

Тоді користувач має записати дані у спеціальну директорію агента, це робиться для того щоб він зумів використовувати введені дані для відправки

листів на конкретну пошту, також необхідно записувати облікові дані в спеціальну директорію, яка має бути обмеженою для всіх користувачів, це можна зробити за допомогою командиpostmap, але відкритою для власниката програм, яким дав дозвіл користувач.

Для інтегрування електронної пошти в платформу Wazuh, знадобиться зайти в конфігураційний файлі із розширенням ossec.configi відредагувати його вміст вказавши такі параметри:

- активацію листів електронної пошти;
- ідентифікатор IP SMTP;
- пошта відправника;
- пошта отримувача яка в нашому випадку так ж сама як і в відправника.

На рисунку 3.10 бачимо вміст файлу із всіма його ключовими параметрами розташованого за посиланням /var/ossec/etc/ossec.conf.

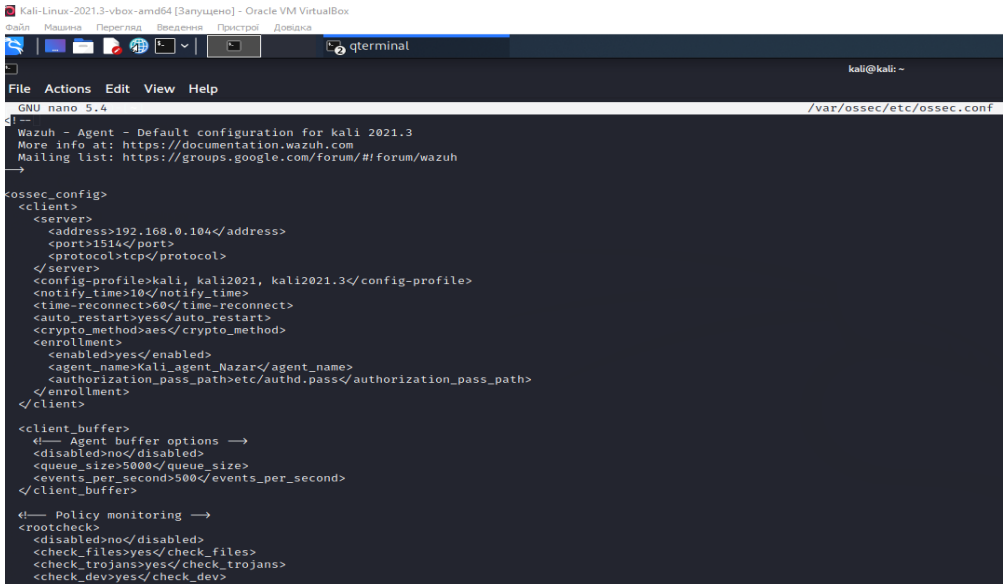


Рисунок 3.10 – Конфігураційний вміст файлу ossec.conf

Перезапустивши служби Wazuh менеджера, ми зможе провірити ефективність роботи налагодженої системи оповіщень. Яка пересилає оброблені дані із інформаційної панелі Wazuh на відповідний канал зв’язку. Таким чином перша шкала вказує на ризик при додаванні незнайомого користувача, а друга

вказуватиме на ризик при включенні незнайомого користувача до локальної групи, що мають права адміністратора.

Загалом Wazuh ефективно ідентифікує дії, які є наслідком атаки шкідливого програмного забезпечення. Однак, для користувача може виявитися складним завданням швидко розпізнати атаку, що триває. Тому корисно мати автоматичні сповіщення, які активуються при виявленні таких ситуацій. Для цього ми використовуємо функціонал оповіщення від OpenDistro (рисунок 3.11), який слугує безкоштовною заміною ElasticWatchers з відкритим вихідним кодом [6].

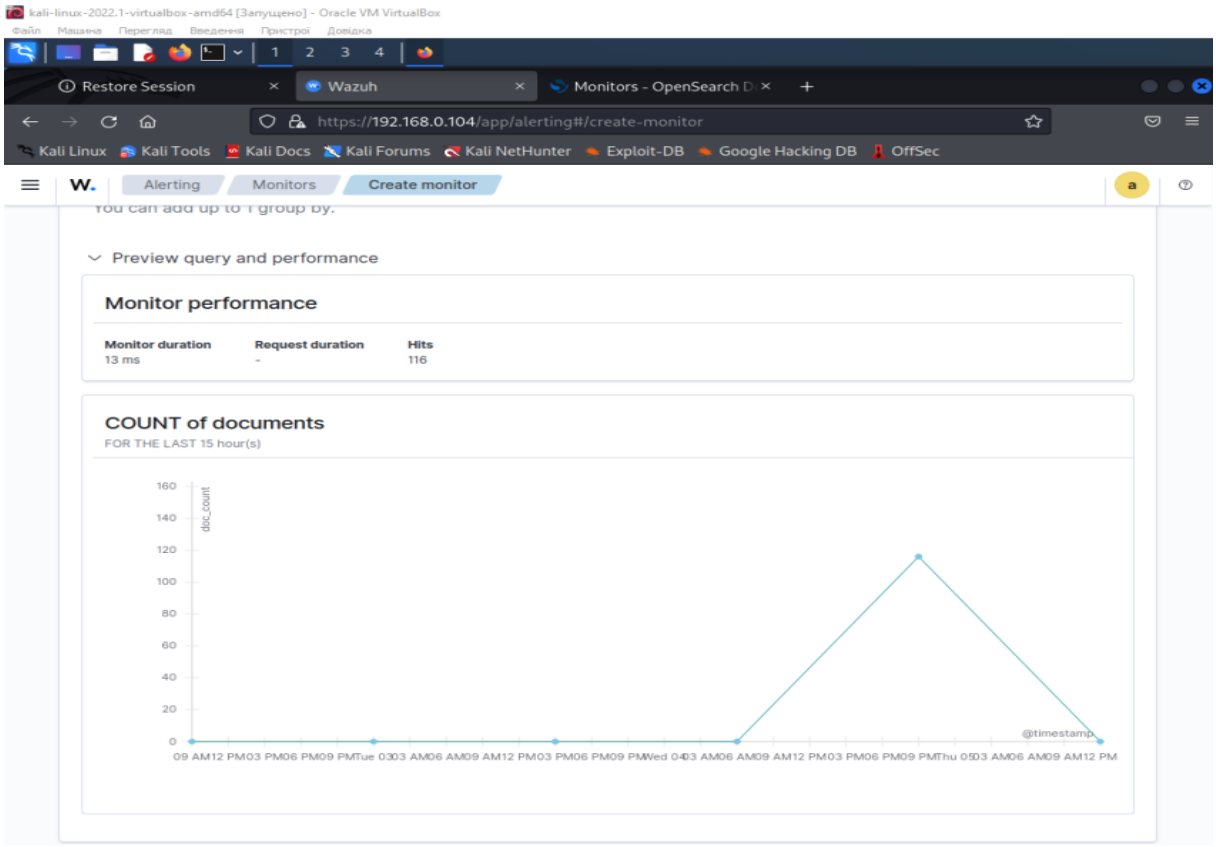


Рисунок 3.11 – Функціонал оповіщень від OpenDistro

На початку заходимо в розділ сповіщень і настроюємо параметри візуалізації нашого монітору, для цього встановлюємо тип моніторингу в нашому випадку він буде представлений у вигляді графіка, далі налаштовуємо формат wazuh-сповіщення який буде задіяний системою, тоді встановлюємо періодичність збору інформації [6].

В процесі налаштування ми отримуємо графік моніторингу, на якому відображатимуться дві шкали верхня показуватиме результати проаналізованих файлів службами Wazuh, а нижня періодичність. А також активовано спеціально призначений для генерації сповіщень при додаванні нових файлів за допомогою фільтра countofdocuments.

Далі ми налаштуємо тригер для монітора (рисунок 3.12). Вводимо ім'я нашого тригера, яке повинно бути повністю унікальним та складатись з літер, чисел, та інших символів. Крім того користувачу надається можливість встановлення персональні вимоги, за якими працюватиме тригер.

В нашому випадку було прописано умову яка описується таким чином, якщо рівень загрози менший 100 то він розглядається як низький, і тоді тригер надсилає список про помічену загрозу на електронну пошту, якщо показник перевалив за 100 тоді тригер на пряму надсилає на робочу область користувача.

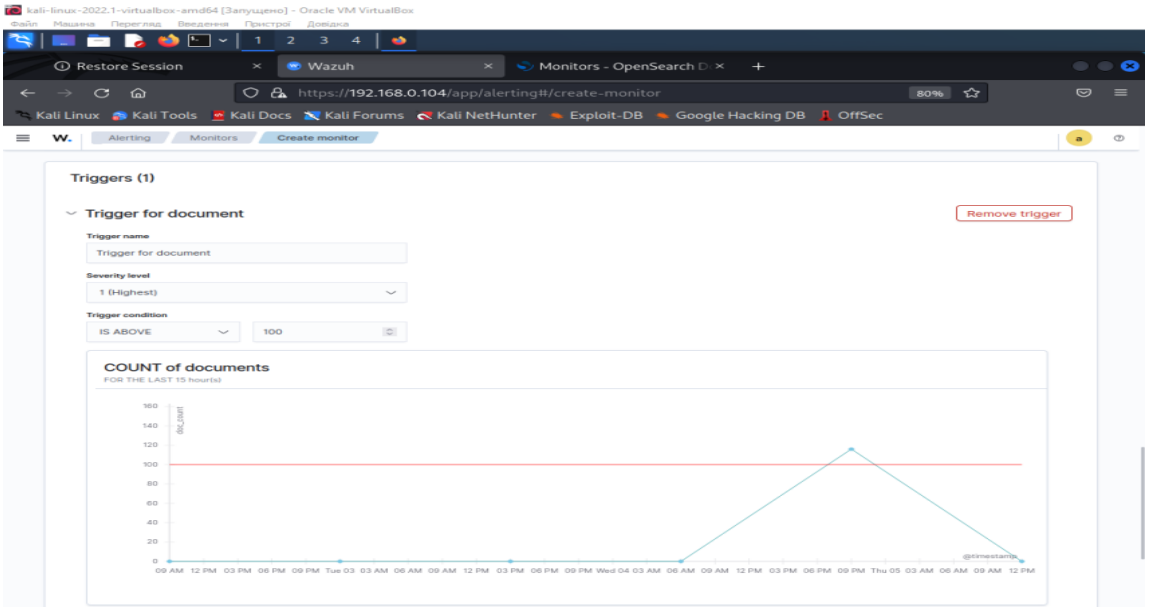


Рисунок 3.12 – Налаштування тригеру для монітора

В результаті ми отримуємо тригер у вигляді графіка із промальованою умовною лінією, яка виконує роль своєрідної планки, що встановлюється згідно чинних вимог користувача. Такий підхід дозволить більш точно показати рівень загрози. Таким чином користувач може змоделювати цілісну картинку про дане шкідливе забезпечення за допомогою поширених методів боротьби включаючи:

аналіз, моніторинг та реагування на інциденти. Крім цього користувач збільшити свою продуктивність, концентруючись на більш важливі процеси.

На противагу першому тригеру що створюється для моніторингу активних файлів було б добре додати тригер що несе інформацію про видалені файли. В цілому тригер для видалених файлів є важливим інструментом, що виявляє ймовірні інциденти, відстежує видалені файли, моніторить поведінку системи та відстежує здійснені дії в ній. Тому наявність додаткового тригеру посилюватиме захист інформаційних ресурсів, та сприятиме ефективному реагуванню на будь яку зловмисну діяльність.

ВИСНОВКИ

В кваліфікаційній роботі було детально викладено принципи роботи програми Wazuh, яка активно займається виявленням та запобіганням вторгненням. Під час виконання роботи було вирішено такі питання.

1. Визначено основні алгоритми роботи систем виявлення та запобігання вторгнень (IDS/IPS). До яких відносять такі ключові системи як: OSSEC, Snort, Suricata та Wazuh.

2. Досліджено архітектуру системи Wazuh. Насамперед звернуто увагу на основні служби досліджуваної системи, а саме Wazuh-indexer, менеджер, та дашборд.

3. Запроваджено заходи безпеки та реалізації методик боротьби із шкідливого програмного забезпечення за допомогою Wazuh. Програмне забезпечення машини було встановлено на віртуальну машину за допомогою загрузочного файлу Ova.

4. Перевірено статуси всіх його конфігуруючих компонентів, а саме менеджера, індексера, та графічної панелі. Перелічені елементи були налаштовані відповідно до вимог зазначених в документації розробника, а також проаналізовані та описані в роботі.

5.Вибрано дві операційні системи Kali Linux. На першу було розгорнуто функціонал програмного продукту Wazuh, а на іншу встановлено агент для подальшого тестування на наявність шкідливого програмного забезпечення, а н в структурі даної операційної системи.

6. Зроблено імітацію шкідливого програмного забезпечення задля наглядного прикладу роботи Wazuh менеджера та усіх його компонентів. Результатом якого є оповіщення про атаку із її детальним описом.

7. Виявлено шкідливе програмне забезпечення та налаштовано тригерну систему моніторингу оповіщень Wazuh.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Malware Statistics for 2024: Trends, Targets and Threat Actors [Електронний режим]. – Режим доступу: https://www.stationx.net/malware-statistics/

2. Illustration of HIDS & NIDS [Електронний режим]. – Режим доступу: https://www.researchgate.net/figure/illustration-of-Network-Intrusion-Detection-System-16_fig7_339297592

3. Коробейнікова Т., Цар О. Аналіз сучасних відкритих систем виявлення та запобігання вторгнень. – Grail of Science

4. Whatis IPS (Intrusion Prevention System)? [Електронний режим]. – Режим доступу: https://www.wallarm.com/what/intrusion-prevention-system

5. Medium: FIM and SIEM with OSSEC [Електронний режим]. – Режим доступу: https://medium.com/tensult/ossec-manager-agents-configuration-dabc6445494f

6. Preventing and detecting ransomware with Wazuh [Електронний ресурс] // Jesus Linares. – 2019. – Режим доступу до ресурсу: https://wazuh.com/blog/preventing-and-detecting-ransomware-with-wazuh/.

7. WINDOWS, LINUX ТА MAC OS — ЧИМ ВІДРІЗНЯЮТЬСЯ ТА ЯКУ ОС ОБРАТИ [Електронний ресурс] // Nspace – Режим доступу до ресурсу: https://www.nspace.ua/info/windows-linux-ta-macos-chim-vidriznyayutsya-ta-yaku-os-obrati.

8. Firch J. HowDoesRansomwareSpread? (5 CommonMethodsIn 2024). *purplesec*.URL: https://purplesec.us/learn/common-ways-ransomware-spreads/.

9. Cloud Intrusion Detection System [Електронний ресурс] // MMR. – 2023. Режим доступу до ресурсу: https://www.maximizemarketresearch.com/market-report/global-cloud-intrusion-detection-system/

10. Whatis a Network Intrusion Detection system (NIDS)? [Електронний ресурс] // Bunny – Режим доступу до ресурсу: https://bunny.net/academy/security/what-is-network-intrusion-detection-nids/.

11. Intrusion Prevention System: What Is An IPS? How Do They Work? [Електронний ресурс] // Okta. – 2023. – Режим доступу до ресурсу: https://www.okta.com/uk/identity-101/intrusion-prevention-system/.

12. Virus Total Documentation Hub [Електронний ресурс] // Virus total – Режим доступу до ресурсу: https://docs.virustotal.com/docs/how-it-works.

13. Download VirtualBox [Електронний ресурс] // VirtualBox – Режим доступу до ресурсу: https://www.virtualbox.org/wiki/Downloads.

14. Choose the desktop you prefer [Електронний ресурс] // Kali – Режим доступу до ресурсу: kali.org

15. Іващенко, М. В. (2023). Система пошуку загроз на основі Wazuh та хмарної платформи Google (Doctoral dissertation, Тернопіль, ЗУНУ).

16. IDS vs. IPS: What Organizations Need to Know [Електронний ресурс] // JosueLedesma. – 2022. – Режим доступу до ресурсу: https://www.varonis.com/blog/ids-vs-ips#ids-types.

17. Getting started with Wazuh [Електронний ресурс] // Wazuh – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/index.html.

18. Ященко, А. М. (2023). Інформаційні технології та інструменти активних дій в кібербезпеці (Master'sthesis, Сумський державний університет).

19. Ковтун, Н. М. (2023). Методи і засоби виявлення вторгнень та інформаційних атак на комп'ютерну систему (Master'sthesis, Тернопільський національний технічний університет імені Івана Пулюя).

20. OSSEC Documentation [Електронний ресурс] // OSSEC – Режим доступу до ресурсу: https://www.ossec.net/docs/index.html.

21. Корченко, А. О. (2019). Методи ідентифікації аномальних станів для систем виявлення вторгнень

22. Лаута, Р. С. (2023). Підвищення стійкості інформаційної безпеки з використанням SIEM-системи Wazuh (Doctoral dissertation, Тернопіль, ЗУНУ).

23. Гавриляк, М. В. (2022). Система виявлення мережевих вторгнень на основі Snort (Doctoraldissertation).

24. Interface Suricata. Linuxboss[Електронне ресурс] – Режим доступу до ресурсу: https://linuxboss.wordpress.com/2015/12/03/graphite-grafana/.

25. Wazuh indexer [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/components/wazuh-indexer.html.

26. Wazuh server [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу:https://documentation.wazuh.com/current/getting-started/components/wazuh-server.html.

27. Agents monitoring and configuration [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html.

28. Data visualization and analysis [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html.

29. Wazuh agent [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу:https://documentation.wazuh.com/current/getting-started/components/wazuh-agent.html.

30. Platform management [Електронний ресурс] // Wazuh. – 2024. – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html.

31. Фляшко Н.Р., Яцків В.В. Алгоритми виявлення шкідливого програмного забезпечення за допомогою Wazuh. ЗБІРНИК НАУКОВИХ ПРАЦЬ за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2024» , Хмельницький, 2024. – С.508-512.

32. Фляшко Н. , Контроль кінцевих точок з використанням агента Wazuh. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2024), Тернопіль, 2024. – С. 18-22.

ДОДАТОК А
Копії публікації

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»

15-16 листопада 2024

Хмельницький 2024

УДК 004.056

Фляшко Н.Р., Яцків В.В.

Західноукраїнський національний університет

АЛГОРИТМИ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ДОПОГОЮ WAZUH

Проведено аналіз можливостей з виявлення шкідливого програмного забезпечення платформи Wazuh. Досліджено основні компоненти, які представляють дану платформу та забезпечують її функціонування. Досліджено алгоритми виявлення шкідливого програмного забезпечення платформи Wazuh.

A research study was undertaken to assess the capabilities of the Wazuh platform in detecting malicious software. The core components of the platform were explored, and the underlying algorithms for malware detection were investigated.

Шкідливе програмне забезпечення становить значну загрозу для людей, фінансів та суспільства в цілому. Його наслідки можуть варіюватися від фінансових втрат до репутаційних збитків для організацій. Шкідливе програмне забезпечення може призвести до крадіжки персональних даних, шахрайських дій та порушень інформаційної безпеки, що призводить до збоїв у роботі системи, уповільнення роботи і навіть виходу з ладу пристроїв. Ці фактори в сукупності сприяють зростанню кількості кібератак, таких як розподілені атаки на відмову в обслуговуванні (DDoS), атаки з вимогами викупу та фішингові схеми. Тому щоб забезпечити захист програмного забезпечення потрібно удосконалювати технічні засоби виявлення шкідливих програм. Wazuh як раз і є тим інструментом, який пропонує доступні рішення для моніторингу мережі та компонентів системи в цілому. А також залучає усі можливі методи та алгоритми виявлення шкідливого програмного забезпечення.

Метою роботи є дослідження алгоритмів виявлення шкідливого програмного забезпечення за допомогою Wazuh та детально ознайомитись із компонентами даної платформи.

Wazuh - це комплексне рішення, яке об'єднує технології системи виявлення вторгнень (IDS) і системи запобігання вторгненням (IPS). Ця платформа розроблена з розширеними можливостями виявлення та реагування (XDR), а також функціями управління інформацією та подіями безпеки (SIEM). Платформа призначена для моніторингу поточних процесів у режимі реального часу, забезпечуючи швидку оцінку діяльності. Крім того, вона може перевіряти цілісність даних, виявляти та запобігати підозрілим діям, а також аналізувати вхідний мережевий трафік на основі попередньо встановлених критеріїв [1].

Wazuh є технічним засобом в основі якого лежать такі елементи: агенти, сервер, панель візуалізації Wazuh dashboard, а також Wazuh indexer.

Wazuh agent є важливим виконавчим механізмом середовища Wazuh. Його основним функціями є збір інформації про стан вузлів, до яких він під'єднаний, а також відправка отриманих результатів на центральний вузол Wazuh server.

Wazuh server – центральний вузол машини Wazuh. Він призначений для збору даних отриманих від агентів та для забезпечення цілісного моніторингу компонентів системи, виявлення шкідливого продукту та реагування на нього [1].В основі серверу лежать вбудовані структури, що відповідають за аналіз загроз. Наприклад структурна компонента MITER ATT&CK попереджатиме про відхилення в мережі посилаючись на встановлені нормативні вимоги. До нормативних вимог належать: HIPPA,GDPR, PCI DSS, та CIS, NIST [1].

Wazuh dashboard – інструмент візуалізації Kibana, який дає змогу в широкому аспекті розширити функціональність системи, дозволяючи користувачам отримати доступ до інтерфейсу із набором функцій [1]. Крім того оброблені дані тестування виводяться на екран та зберігаються в платформі, а також інтерфейс надає можливості створення таблиць, діаграм, та індивідуальних звітів за допомогою вбудованих інструментів. .

Wazuh indexer – використовується для інтеграції із різними системами моніторингу, розподіляючи дані формату JSON між сегментами, тим самим формуючи зв'язок, що об'єднується в вузли [1]. Такий підхід може створити певну надлишковість запитів що формуються між вузлами. Однак, така взаємодія покращуватиме продуктивність роботи та збільшуватиме пропускну здатність запитів у системі. В структурі індексатора розглядають такі чотири категорії: оповіщення Wazuh, архівна складова, моніторинг та статистичні дані. Кожен підтип має свою зону відповідальності із вузько спеціалізованими технічними рішеннями.

В процесі аналізу шкідливого програмного забезпечення система Wazuh використовує різні алгоритми роботи.

Алгоритм заснований на основі сигнатур (рисунок 1). Цей алгоритм пропонує користувачам шаблонний підхід виявлення шкідливого програмного забезпечення, який базується на використанні бібліотек, що складаються із ідентифікованих зловмисних програм [2]. В процесі сканування файл порівнюється із шаблоном, що є в бібліотеці та визначає чи він збігається з будь-яким із цих шаблонів, якщо перевірка встановила ідентичність із будь яким шаблоном то він позначатиметься як загроза. Цей процес має такі самі риси, як і традиційне антивірусне програмне забезпечення - ідентифікуючи загальновідомі віруси за їхніми сигнатурним вмістом. Для того щоб даний метод зумів виявляти більше шкідливих програм рекомендується постійно оновлювати базу даних сигнатур, щоб вона могла швидко розпізнавати та усувати будь-які потенційні загрози.

Алгоритм виявлення аномалій розроблений для фіксації будь якої підозрілої активності в поведінці системи. На рисунку 2 зображено блок схему алгоритму роботи аномалій IDS. Безперервне відстеження активностей в системі та

регулярне порівняння поведінок дозволить точно відслідкувати будь-які відхилення, і таким чином встановити потенційні загрози. Наприклад, якщо було виявлено аномальний обсяг трафіку [2], це може свідчити про потенційне вторгнення.

Рисунок 1 – Алгоритм роботи сигнатурного методу

Цей підхід особливо ефективний для виявлення експлойтів нульового дня, які часто пов'язані з аномально високим рівнем трафіку. Проте даний підхід має і свої недоліки, в процесі роботи користувач може потребувати більше затрат ресурсів мережі, тим самим збільшуючи її трафік, а тому система зосереджується на ту активність яка є безпечною, замість того щоб виявляти шкідливу активність. Щоб звести до мінімуму помилкові тривоги, важливо правильно налаштувати тригери в IDS-системі, щоб вона сповіщала лише про справжні аномалії, які можуть виникнути.

Також Wazuh платформа здійснює моніторинг журналів подій будь яких джерел а саме: застосунків, операційних систем, чи мережевих пристроїв, виявляє вразливості без участі користувача та блокує їх, забезпечує централізоване управління, підтримує можливість інтеграції із іншими технічними ресурсами, що відповідають за безпеку.

Рисунок 2 – Блок-схема алгоритму роботи на основі аномалій

Проведений аналіз алгоритмів виявлення шкідливого програмного забезпечення за допомогою Wazuh показали, що дана система є ефективним та надійним засобом для виявлення зловмисних програм та формування відповіді на інциденти в процесі роботи. Вона наділена ешелонованою архітектурою та різноманітними функціональними можливостями, що дозволяє запроваджувати її модель в структури організацій, які прагнуть захистити свої активи.

Перелік посилань.

1. Architecture of the wazuh system [Електронний ресурс] // Wazuh documention. – 2024. – Режим доступу до ресурсу: https://documentation.wazuh.com/current/getting-started/components/index.html.
2. Коробейнікова Т., Цар О. Аналіз сучасних відкритих систем виявлення та запобігання вторгнень. Міжнародний науковий журнал «Грааль науки», |No 27 (травень, 2023) – С. 317-325



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В'ЯЧЕСЛАВА ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2024)

науково-практична конференція
молодих вчених, аспірантів та студентів

26–28 серпня 2024
Тернопіль

ЗМІСТ

СИСТЕМИ ТА ТЕХНОЛОГІЇ КІБЕРБЕЗПЕКИ

УДК 004.056

Назарій ФЛЯШКО

Західноукраїнський національний університет

КОНТРОЛЬ КІНЦЕВИХ ТОЧОК З ВИКОРИСТАННЯМ АГЕНТА WAZUH

Вступ. Як відомо шкідливе програмне забезпечення широко розповсюдження в кібер просторі. Тому компанії щоб вберегтися та застерегти себе від дії зловмисного програмного забезпечення використовують різноманітні продукти які виявляють, повідомляють або ж усувають їх без участі користувача. Одним із таких ресурсів є платформа Wazuh. Успішне виконання задач та надійний захист програмних ресурсів та активів дозволила їй завоювати схвалення та прихильність сотні тисяч компаній. Крім цього Wazuh наділений багатьма корисними функціями, включаючи легку інтеграцію із моделями SOC, активне реагування на інциденти та шкідливе програмне забезпечення, постійний моніторинг пристроїв, та їх цілісна оцінка безпеки та пошуку вразливих зон, можливість працювати з великими даними локальних чи хмарних середовищах, ідентифікація прихованих процесів. Головним виконавчим механізмом, що забезпечує комунікацію між платформою та серверними, комп'ютерними і мережевими точками є агент. Агент займається збором, аналізом, моніторингом, виявленням, реагуванням шкідливого програмного забезпечення із централізованим управлінням.

Мета: дослідити виявлення шкідливого програмного забезпечення за допомогою агента, що є центральним компонентом платформи Wazuh, а також вивчити його структуру та технічні особливості.

1. Архітектура агента Wazuh

Агент є центральним елементом в структурі Wazuh машини, адже в нього вбудовано багато модулів, без яких дана система не працювала б. Робоча область агента поширена на вузлових точках серверного, користувальницького та мережевого середовища. У процесі встановлення завантажується програмне забезпечення агента, після чого налаштовуються його процеси під певний тип задач, які він має проробити.

За своєю архітектурою агент має модульну будову. Вузька направленість кожного елементу дозволяє ефективно вирішувати певні задачі, тобто кожний метод розробляється під конкретні завдання. Таким чином агент здатний відстежувати дані файлової системи, зчитувати повідомлення із журналів, збирати об'єктивну інформацію про наявність шкідливого програного забезпечення.

На рисунку 1 наведено архітектуру агента Wazuh із його основними модулями.

Модуль збору журналів використовується для зчитування файдів журналів та події що відбуваються в операційних системах. Конфігураційний файл потрібний для зручного редагування системного коду агента, пристосовуючи його під власні потреби.

Комунікаційний модуль сполучає агент із менеджером Wazuh, де отримані

результати роботи агента направляються на сервер Wazuh для подальшого аналізу.

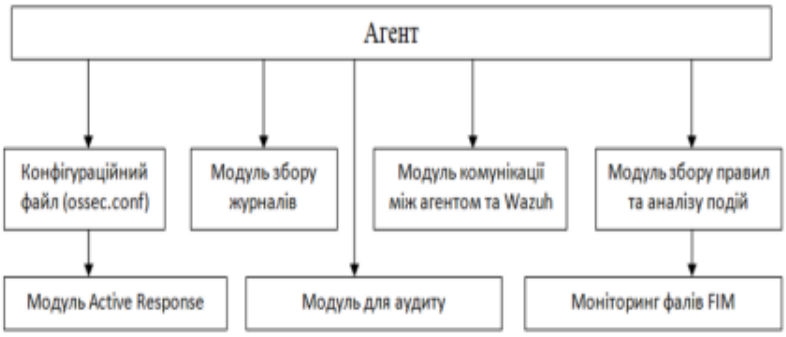


Рисунок 1- Архітектурна схема агента Wazuh

Моніторинг файлової систем більш відомий як FIM розроблений для введення спостереження за файловою системою, що дозволяє в режимі реального часу активно відслідковувати будь які зміни у файлах, а також перевіряти їх попередня стани для віддалених запитів.

Модуль збору правил та аналізу подій займається скануванням та збором даних, включаючи версії операційної системи, інформацію про мережу, відомості про запущені процеси, завантажені програми, а також санують порти що перебувають в активному режимі.

Модуль аудиту використовується для забезпечення цілісного аналізу досліджуваного об'єкту, а саме операційної системи, тобто розробляються усі можливі рішення для виявлення та захисту операційних систем від впливу шкідливого продукту.

Активне реагування розроблене для автоматичного виявлення загроз, що були помічені, блокуючи мережевий трафік, та інші процеси, видаляючи шкідливі файли [2].

Після успішного розгортання інтерфейсу користувача наступним кроком буде відвідування офіційного сайту розробників. Тут ви знайдете детальні інструкції щодо встановлення моделей агентів для певного типу операційної системи. У нашому випадку ми будемо встановлювати програмне забезпечення агента на операційну систему Kali Linux. Існує два способи встановлення агента. Перший спосіб передбачає використання набору команд у терміналі, як зазначено в документації. Другий спосіб передбачає безпосереднє введення інформації про агента та пристрій [1]. Розглянемо обидва способи докладніше.

Якщо ми вирішили налаштувати його через консоль, то після встановлення відкриється інтерфейс агента. Потім нам потрібно призначити агенту IP-адресу менеджера Wazuh (ту саму адресу, що використовується для запуску програмного забезпечення) і ввести ключ для входу.

Крім того, якщо ми вирішимо заповнити форму, доступну за посиланням Додати агента, нам потрібно надати інформацію про агента, як показано на рисунку 2.

Після заповнення форми буде згенеровано команду [1]. Цю команду потрібно виконати в командному рядку, щоб запустити агента з усіма його функціональними можливостями.

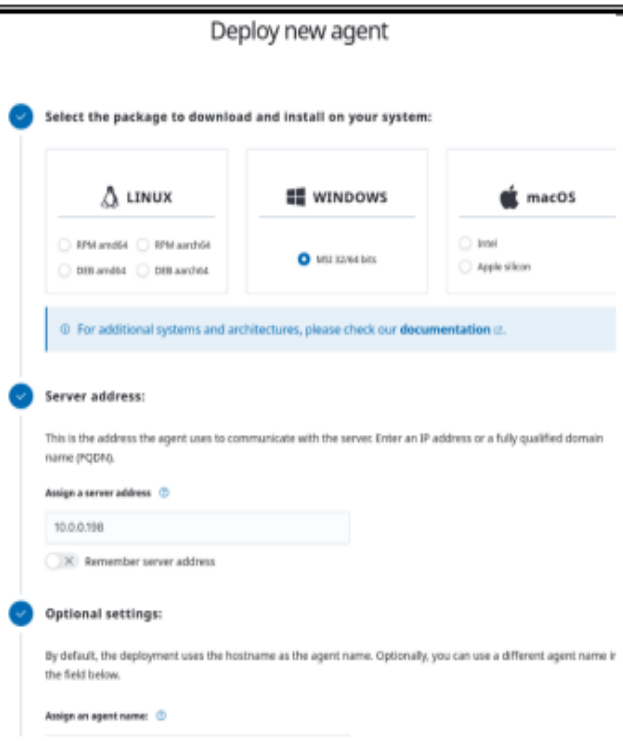


Рисунок 2 - Ручне заповнення форми агента

Тоді коли ми внесли усі необхідні відомості у форму агента, в інтерфейсі користувача з'явиться вкладка активно запущеного агента. Корисним є те що за потреби ми можемо в любий час налаштовувати його параметри, та переглядати інформацію, яку йому вдалося зібрати із системного середовища до якого він прив'язаний за IP адресом.

Як ми бачимо на малюнку показано конфігурацію агента з його важливими даними. Тоді коли ми внесли усі необхідні відомості у форму агента, в інтерфейсі користувача з'явиться вкладка активно запущеного агента. Корисним є те що за потреби ми можемо в любий час налаштовувати його параметри, та переглядати інформацію, яку йому вдалося зібрати із середовища до якого він прив'язаний за IP адресом.

В цю панель можливо підключити велику кількість агентів, а отже користувач зможе активно прив'язуватись до операційних систем та досліджувати їхню поведінку. Від агентів користувач отримує повну звітність активних подій, які виконувались системою вході яких користувач формуватиме висновки щодо посилення безпеки.

Щоб прив'язати агента Wazuh до каталогу Linux, ми повинні спочатку встановити та налаштувати агента Wazuh на основну операційну систему. Після цього необхідно виконати наведені нижче кроки, для того щоб прив'язати агента до конкретного файлу в каталозі Linux:

1. Відкрити оболонку термінал в системі Linux.
2. Перейти до відповідного каталогу, в якому встановлено агент Wazuh.
3. Виконайте команду прив'язки агента до цього каталогу, вказавши необхідні параметри, такі як ім'я сервера або IP-адреса сервера керування Wazuh.

Встановивши зв'язок між агентом і кожним елементом каталогу, користувач зможе отримати вичерпну інформацію про кожен файл у каталозі та визначити всі наявні в ньому вразливості. Дані будуть зібрані та належним чином надіслані на сервер Wazuh.

Для того, щоб проілюструвати поведінку сервера Wazuh у випадку атаки зловмисників, необхідно написати в терміналі наступну функцію: У терміналі слід ввести наступну команду [3]:

python3 wazuh-ransomware-poc.py attack.

Після виконання цієї команди почнеться процес дублювання файлів, в результаті якого існуючі файли будуть видалені, а нові зашифровані [3].

Згодом на інформаційній панелі Wazuh з'явиться повідомлення про виявлення подій, що відбулися під час моделювання цієї атаки. В інтерфейсі користувача показаного на рисунок 3 ми спостерігатимемо за моніторингом файлової системи, який відображатиме зображення перезапису файлів і бачитимемо деякі відмінності із попереднім рисунком, які виділені кольором що сигналізують про додавання "added" та видалення "deleted" оригінальних файлів [3].

full_log	rule.description	syscheck.event
File '/home/vagrant/test/Directory_09/File_18.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_19.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_17.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_17.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_16.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_15.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_14.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_11.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_11.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_08.txt' was deleted.	File deleted.	deleted
File '/home/vagrant/test/Directory_09/File_07.txt.encrypted' was added.	File added to the system.	added
File '/home/vagrant/test/Directory_09/File_06.txt.encrypted' was added.	File added to the system.	added

Рисунок 3 - Моніторинг файлової системи після атаки програм-вимагачів

Висновок. Проведені дослідження алгоритмів виявлення шкідливого програмного забезпечення показали ефективність агента на платформі Wazuh. Було досліджено архітектуру агента, налаштовано його конфігурацію, а також показано його роботу на ділі.

Перелік використаних джерел.

1. Instalation Wazuh agent [Електронний ресурс]. - Режим доступу: https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html

2. Active response [Електронний ресурс]. - Режим доступу: https://documentation.wazuh.com/current/user-manual/capabilities/active-response/index.html

3. Preventing and detecting ransomware with Wazuh [Електронний ресурс] // Jesus Linares. - 2019. - Режим доступу до ресурсу: https://wazuh.com/blog/preventing-and-detecting-ransomware-with-wazuh/.