МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

Методичні рекомендації до виконання лабораторних робіт
з дисципліни

МОНІТОРИНГ ТА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ

для здобувачів освітньо-професійної програми
«Кібербезпека»
другого (магістерського) рівня вищої освіти
спеціальності «Кібербезпека та захист інформації»

Тернопіль – ЗУНУ
2025

Методичні рекомендації до виконання лабораторних робіт з дисципліни «Моніторинг та управління інформаційною безпекою» для здобувачів освітньо-професійної програми «Кібербезпека» другого (магістерського) рівня вищої освіти спеціальності «Кібербезпека та захист інформації» / Укл.: Касянчук М.М., Івасьєв С.В., Нетребяк М. Р. – Тернопіль: ЗУНУ, 2025. – 29 с.

Відповідальний за випуск: Івасьєв С.В., к.т.н., доцент кафедри кібербезпеки

Рецензенти:

Загородна Н.В. к.т.н., доцент, завідувач кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя

Батько Ю.М. к.т.н., доцент кафедри комп'ютерної інженерії факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету

Методичні вказівки розглянуті та схвалені на засіданні кафедри кібербезпеки, протокол № 10 від 24 квітня 2025 р.

ЛАБОРАТОРНА РОБОТА №1

Тема: Оцінка ризиків інформаційної безпеки

Мета роботи: Метою цієї лабораторної роботи є ознайомлення студентів з методами оцінки ризиків для організаційних систем інформаційної безпеки. Студенти навчаться застосовувати методи оцінки ризиків, такі як OCTAVE, NIST SP 800-30 або ISO/IEC 27005, для ідентифікації, оцінки та управління інформаційними ризиками.

Необхідні інструменти:

- **OpenVAS** – система для сканування вразливостей.
- **RiskWatch** – програмне забезпечення для оцінки ризиків.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Оцінка ризиків є важливим елементом управління інформаційною безпекою, оскільки вона дозволяє виявити та оцінити потенційні загрози для інформаційних активів. Оцінка ризиків включає кілька етапів:

- ідентифікація ризиків – виявлення загроз для інформаційних активів.,
- оцінка ймовірності їх виникнення – визначення частоти та ймовірності атак,
- визначення впливу – оцінка можливих наслідків загроз,
- прийняття рішень щодо управління ними – вибір методів усунення або зменшення ризиків.

Відомі методи оцінки ризиків включають:

- OCTAVE: оцінка ризиків в рамках організаційного контексту. Включає три фази: побудова профілю загроз, ідентифікація вразливостей та аналіз та визначення стратегії управління ризиками. **OCTAVE Framework:** https://www.cert.org/octave
- ISO/IEC 27005: міжнародний стандарт для управління ризиками. Описує процес оцінки ризиків, включаючи ідентифікацію, аналіз та управління ризиками **ISO/IEC 27005 Overview:** https://www.iso.org/standard/75281.html
- NIST SP 800-30: посібник з оцінки ризиків у державних установах США. Включає аналіз активів, загроз, вразливостей та впливу на систему. **NIST SP 800-30:** https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

ПОРЯДОК ВИКОНАННЯ РОБОТИ

Завдання:

- Ознайомлення з методами оцінки ризиків.
- Налаштування OpenVAS для сканування вразливостей.
- Оцінка ризиків та аналіз отриманих результатів.
- Розробка стратегії управління ризиками.

Завдання 1: Ознайомлення з методами оцінки ризиків

Крок 1: Ознайомтесь із методологією оцінки ризиків OCTAVE, NIST SP 800-30 та ISO/IEC 27005.

Крок 2: Знайдіть інформацію про особливості кожного методу та їх використання в оцінці ризиків.

Крок 3: Підготуйте коротке пояснення (0,5–1 стор.) щодо застосування обраного методу. Опишіть у звіті основні етапи вибраної методики.

Завдання 2: Налаштування OpenVAS

Крок 1: Завантажте та встановіть **OpenVAS** на локальну систему або віртуальну машину.

1. OpenVAS включений у репозиторії **Kali Linux**, тому встановлення виконується командами:
   ```
   sudo apt update
   sudo apt upgrade
   sudo apt install gvm -y  / sudo apt install -y openvas
   ```
2. Після встановлення запустіть ініціалізацію та оновлення бази вразливостей:
   ```
   sudo gvm-setup
   ```

Переконайся, що OpenVAS повністю оновлений:

```
sudo gvm-feed-update
```

Перевірте статус сервісів:

```
sudo gvm-check-setup
```

Якщо є помилки, виправте їх згідно з підказками в терміналі.

Крок 2: Запустіть службу:

```
sudo gvm-start
```

Перевірка статусу. Якщо все добре, відобразиться повідомлення:

"It seems like your GVM-Setup is OK"

```
        WARNING: Your password policy is empty.
        SUGGEST: Edit the /etc/gvm/pwpolicy.conf file to set a password policy.
Step 9: Checking greenbone-security-assistant ...
        OK: greenbone-security-assistant is installed

It seems like your GVM-23.11.0 installation is OK.
```

Встановлення OpenVAS на **Ubuntu / Debian**

1. Оновіть систему та встановіть необхідні пакети:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y software-properties-common curl
```

2. Додайте репозиторій Greenbone:

```
sudo add-apt-repository ppa:mrazavi/gvm
sudo apt update
```

3. Встановіть OpenVAS:

```
sudo apt install -y gvm
```

4. Ініціалізація та оновлення бази вразливостей.
5. Запуск сервісу.

Крок 3: Виконання сканування мережі для виявлення вразливостей. Доступ до веб-інтерфейсу GVM (OpenVAS)

- Відкрийте браузер і перейдіть за адресою: https://127.0.0.1:9392

- Логін: admin
- Пароль можна знайти за допомогою команди:

```
sudo cat /var/lib/gvm/users/admin/password
```

Перевірка існування адміністратора:

```
sudo runuser -u _gvm -- gvmd --get-users
```

Створення нового адміністратора

```
sudo runuser -u _gvm -- gvmd --create-user=admin --password=Пароль
```

Перевірка створення адміністратора:

```
sudo runuser -u _gvm -- gvmd --get-users
```
Якщо пароль не вдаэться дызнатись, можна його змінити командою:
```
sudo runuser -u _gvm -- gvmd --user=admin --new-password=Новий_пароль
```

Завдання 2: Налаштування OpenVAS для сканування вразливостей.
Крок 1: Створення цілі:

- Створіть створіть профіль для сканування мережі. Перейдіть у **"Configuration" → "Targets"**
- Додайте IP-адресу або діапазон IP-адрес, які необхідно скануватия.
- Збережіть.

Крок 2: Запуск сканування:

- Виконайте сканування заданої мережі (локальної або тестової). Перейдіть у **"Scans" → "Tasks"**
- Створіть нове завдання (Task).
- Виберіть створену ціль
- Запустіть процес сканування, натисніть **Start** та дочекайтеся завершення**.**
- Запишіть виявлені вразливості та проаналізуйте їх рівень загрози.

Завдання 3: Оцінка ризиків та аналіз отриманих результатів

Крок 1: Проаналізуйте виявлені вразливості та визначте ймовірні загрози. Оцініть ймовірність експлуатації кожної знайденої вразливості. Класифікуйте знайдені загрози за рівнем критичності: критичні (Critical), високі (High), середні (Medium), низькі (Low). Визначте найбільш критичні вразливості у системі.

Крок 2: Використовуючи RiskWatch, оцініть рівень ризиків для ваших активів, враховуючи ймовірність і наслідки кожної загрози.

1. Встановіть програмне забезпечення RiskWatch або скористайтеся альтернативним методом оцінки ризиків.
2. Використовуйте наступні критерії оцінки ризиків: ймовірність загрози (наприклад, на основі історії атак), вплив загрози (наслідки в разі реалізації атаки).
3. Проведіть аналіз та визначте рівень ризику для кожної знайденої вразливості.
4. Опишіть у звіті основні ризики та можливі наслідки.

Завдання 4: Розробка стратегії управління ризиками

1. Запропонуйте заходи щодо усунення або мінімізації ризиків (патчі, зміна конфігурацій тощо).
2. Опишіть дії щодо зменшення впливу загроз (резервне копіювання, моніторинг).
3. Запропонуйте план управління ризиками, який включає опис загроз та їх потенційного впливу, запропоновані заходи захисту та пом'якшення виявлених загроз, оцінку залишкового ризику після реалізації заходів.

Виконання лабораторної передбачає **три рівні складності** – базовий, середній та високий, що дозволить підвищувати свої навички в оцінці ризиків інформаційної безпеки..

Рівень	Що потрібно зробити	Зміст звіту
базовий (60-74 балів)	Ознайомтесь із методологією оцінки ризиків OCTAVE, NIST SP 800-30 та ISO/IEC 27005.	Опишіть основні етапи вибраної методики та поясніть їх роль у процесі оцінки ризиків.
середній (75-89 балів)	**Виконати всі дії базового рівня.** Встановити та налаштувати OpenVAS. Виконати сканування локальної мережі на наявність вразливостей. Проаналізувати отримані результати.	Все з базового рівня. Опис інсталяції та налаштування OpenVAS. Результати (знімки екрану) виконаного сканування. Перелік виявлених вразливостей з їх описом та потенційним впливом.

		Аналіз критичних загроз і рекомендації щодо їх усунення.
високий (90-100 балів)	**Виконати всі дії середнього рівня.** Використовуючи RiskWatch, оцінити рівень ризиків для виявлених загроз. Проаналізувати ймовірність реалізації атак через виявлені вразливості. Запропонувати базові заходи зниження ризиків.	Все з середнього рівня. Аналіз ризиків, використовуючи RiskWatch. Визначення потенційних загроз та їх впливу на систему. Загальна оцінка ризиків у тестованій системі. Рекомендовані заходи зниження ризиків, включаючи технічні, організаційні та адміністративні методи.

Контрольні питання

1. Які методи оцінки ризиків існують і в чому їхні відмінності?
2. Які етапи включає процес оцінки ризиків в організації?
3. Як оцінюються наслідки вразливостей у процесі аналізу ризиків?
4. Що таке ризик інформаційної безпеки? З яких компонентів він складається?
5. Які основні етапи процесу управління ризиками інформаційної безпеки?
6. У чому полягає різниця між загрозою, вразливістю та ризиком?
7. Які методології оцінки ризиків інформаційної безпеки ви знаєте?
8. Коротко охарактеризуйте методологію **OCTAVE**: які основні кроки вона передбачає?
9. У чому особливості методології **NIST SP 800-30** при оцінці ризиків?
10. Яку роль відіграє стандарт **ISO/IEC 27005** у процесі управління ризиками?
11. Які ключові показники використовуються для оцінки рівня ризику?
12. Що таке матриця ризиків і як вона допомагає у прийнятті рішень?
13. Для чого використовується система **OpenVAS** і які дані вона надає при оцінці ризиків?
14. Які основні можливості програмного забезпечення **RiskWatch**?
15. Як результати сканування вразливостей (наприклад, з OpenVAS) інтегруються у процес оцінки ризиків?
16. Які дії можна виконати після виявлення високого рівня ризику?
17. У чому різниця між кількісною та якісною оцінкою ризиків?
18. Наведіть приклади заходів для зниження ризиків у сфері інформаційної безпеки.

ЛАБОРАТОРНА РОБОТА №2

Тема: Встановлення клієнта Wazuh на Ubuntu та підключення його до сервера моніторингу безпеки.

Мета роботи: Ознайомитися з процесом встановлення та налаштування клієнтського агента Wazuh на операційній системі Ubuntu. Навчитися підключати клієнта до вже встановленого сервера Wazuh для збору та централізованого аналізу даних безпеки.

Необхідні інструменти:

1. **Операційна система**: Ubuntu (версія 20.04 або новіша).
2. **Wazuh agent –** клієнтський агент для збору логів та подій.
3. **Доступ до Wazuh Manager (сервера) –** попередньо налаштований сервер з Wazuh.
4. Права адміністратора **(sudo)** на клієнтській машині.
5. Термінал / **SSH-клієнт** (наприклад, OpenSSH) для встановлення та налаштування.
6. **Systemctl –** для керування сервісами агента**.**

ТЕОРЕТИЧНІ ВІДОМОСТІ

Wazuh — це відкрита платформа для моніторингу безпеки та управління подіями, яка дозволяє виявляти загрози, аналізувати журнали подій та контролювати стан інформаційних систем. Вона поєднує функціонал системи виявлення вторгнень, аналізатора логів, моніторингу цілісності файлів та засобів для виявлення вразливостей.

Архітектура Wazuh складається з центрального сервера, що приймає та аналізує дані, та клієнтських агентів, які встановлюються на робочі станції чи сервери й збирають інформацію про їхній стан. Для зручної роботи з даними платформа інтегрується з Elastic Stack і надає візуалізацію через панель Kibana.

Завдяки широким можливостям Wazuh використовується для виявлення атак, управління ризиками та забезпечення відповідності стандартам безпеки. Це робить систему ефективним інструментом у сфері кібербезпеки для організацій різного масштабу.

ПОРЯДОК ВИКОНАННЯ

Завдання:

- Встановлення клієнта **Wazuh agent**.
- Налаштування **Wazuh agent** та підключення до сервера.

Завдання1: Встановлення клієнта **Wazuh agent**.

Крок 1. Оновлення системи

sudo apt update && sudo apt upgrade -y

Крок 2. Встановлення агента Wazuh

Завантажуємо інсталяційний скрипт:

curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh

Запускаємо з правами адміністратора:

sudo bash wazuh-install.sh --install-agent

Крок 3. Налаштування клієнта

Файл конфігурації агента:

sudo nano /var/ossec/etc/ossec.conf

У секції <client> вказати IP-адресу сервера Wazuh Manager:

```
<client>
  <server>
    <address>10.65.0.106</address>
    <port>1514</port>
    <protocol>tcp</protocol>
  </server>
</client>
```

Зберегти зміни та закрити файл.

Крок 4. Реєстрація клієнта на сервері (можна здійснити через веб-інтерфейс, як вказано в завданні 2)

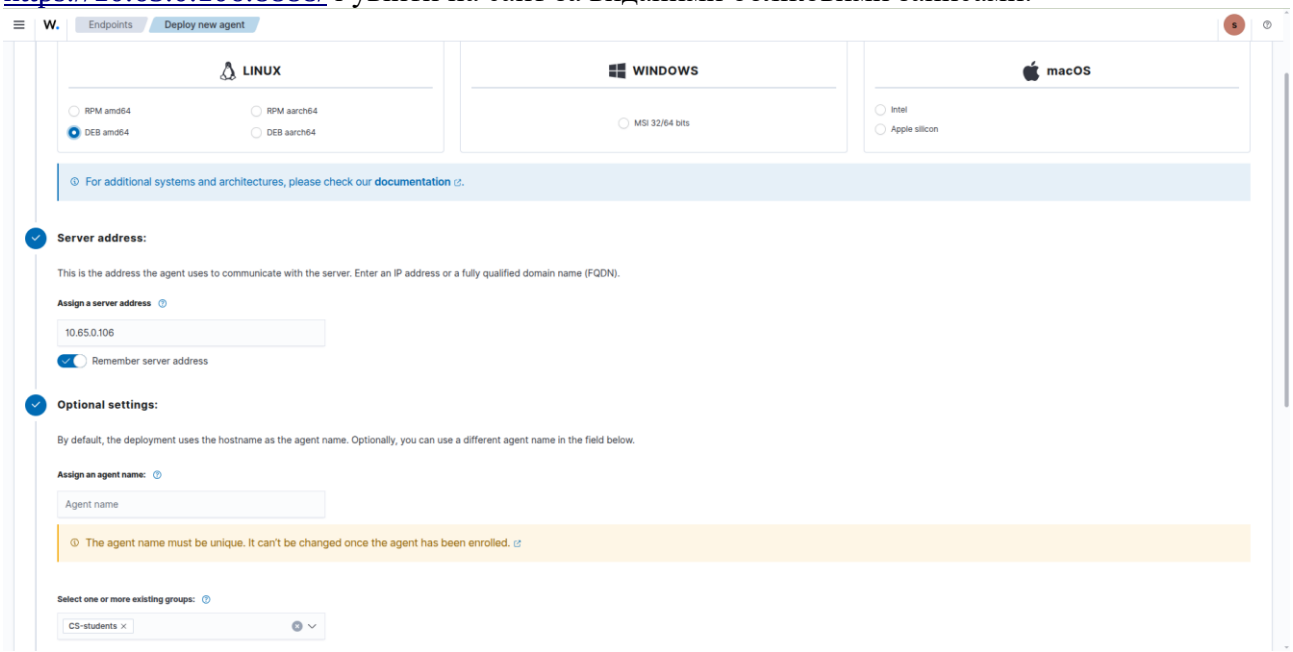
На сервері Wazuh Manager:

sudo /var/ossec/bin/manage_agents

Вибрати опцію (A) Add an agent.
Ввести ім’я, IP-адресу клієнта та згенерований ключ.
Скопіювати отриманий ключ (його можна отримати виконавши завдання 2).
На клієнті:
sudo /var/ossec/bin/manage_agents
Вибрати (I) Import key from the manager.
Вставити скопійований ключ.
Крок 5. Запуск агента
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Перевірка статусу:
sudo systemctl status wazuh-agent
Крок 6. Перевірка з’єднання з сервером
На клієнті:
sudo tail -f /var/ossec/logs/ossec.log
Повинні з’явитися повідомлення про успішне підключення до сервера.
На сервері:
sudo /var/ossec/bin/agent_control -l
У списку повинен відображатися новий клієнт.

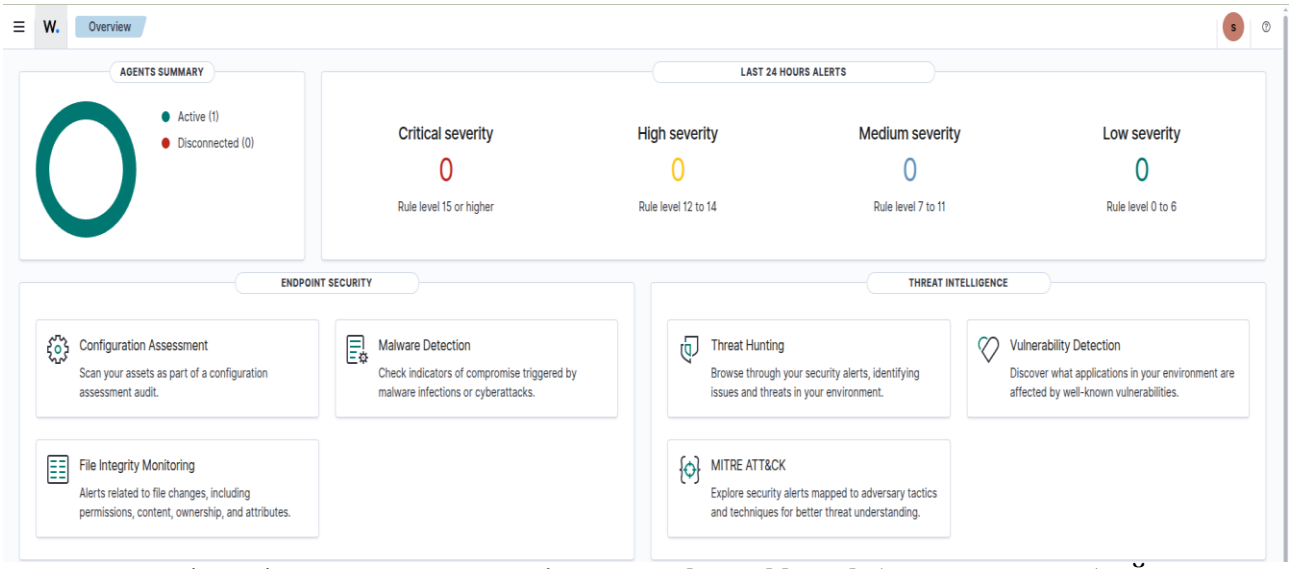
Завдання2: Налаштування **Wazuh agent** та підключення до сервера.

Для виконання завдання необхідно скористатись цифровою інфраструктурою навчальної лабораторії тільки у **ЛОКАЛЬНІЙ** мережі ЗУНУ за посиланням: https://10.65.0.106:8888/ і увійти на сайт за виданими обліковими записами.



Далі на вибір можна додати власний хост, власну віртуальну машину, або скористатись уже наявною віртуальною машиною для виправлення вразливостей.

Щоб додати власну машину до панелі, після входу у панель Wazuh потрібно перейти у випадаючому меню на Agents Management і ввести всі необхідні дані, додавши машину до групи вузлів — CS-students.



На скріншоті показано головне вікно **Wazuh Dashboard** (оглядова панель). Його можна розділити на кілька основних розділів:

1. Agents Summary (Підсумок агентів)

У верхньому лівому блоці відображається кількість агентів, які підключені до сервера Wazuh. Вказується, скільки з них Active (активні) і Disconnected (відключені). Це дозволяє швидко зрозуміти, які клієнтські машини успішно надсилають дані, а які потребують перевірки.

2. Last 24 Hours Alerts (Оповіщення за останні 24 години)

У верхньому правому блоці показано кількість подій (alerts) різної критичності:

Critical severity – критичні (найнебезпечніші, рівень правил 15 і вище),

High severity – високої небезпеки (рівень правил 12–14),

Medium severity – середньої небезпеки (рівень правил 7–11),

Low severity – низької небезпеки (рівень правил 0–6).

Цей блок допомагає швидко оцінити безпекову ситуацію.

3. Endpoint Security (Безпека кінцевих точок)

У нижньому лівому секторі відображаються модулі, які контролюють стан клієнтських систем:

Configuration Assessment – перевірка конфігурацій та відповідності стандартам.

Malware Detection – виявлення шкідливих програм та компрометацій.

File Integrity Monitoring – контроль цілісності файлів (зміни у вмісті, правах доступу, власниках тощо).

4. Threat Intelligence (Інформація про загрози)

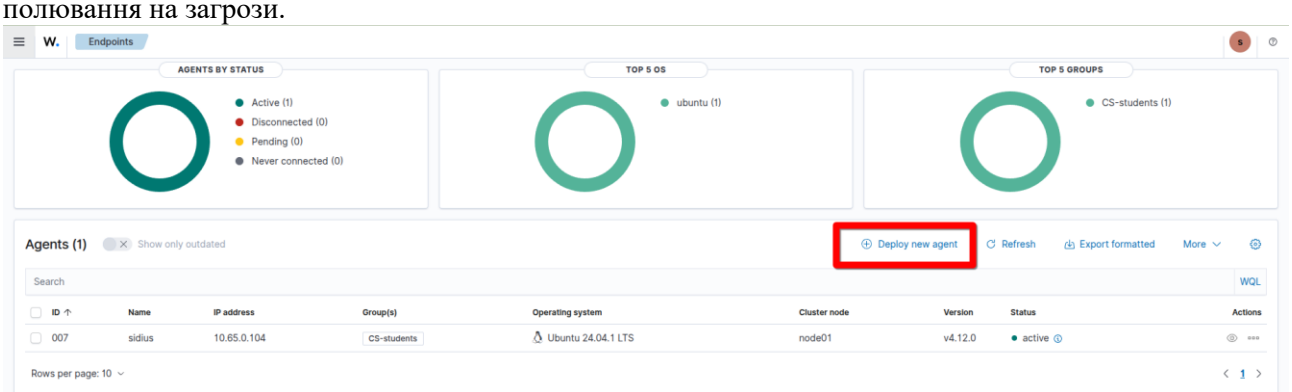
У нижньому правому секторі представлені інструменти для аналізу атак та вразливостей:

Threat Hunting – пошук загроз у зібраних даних.

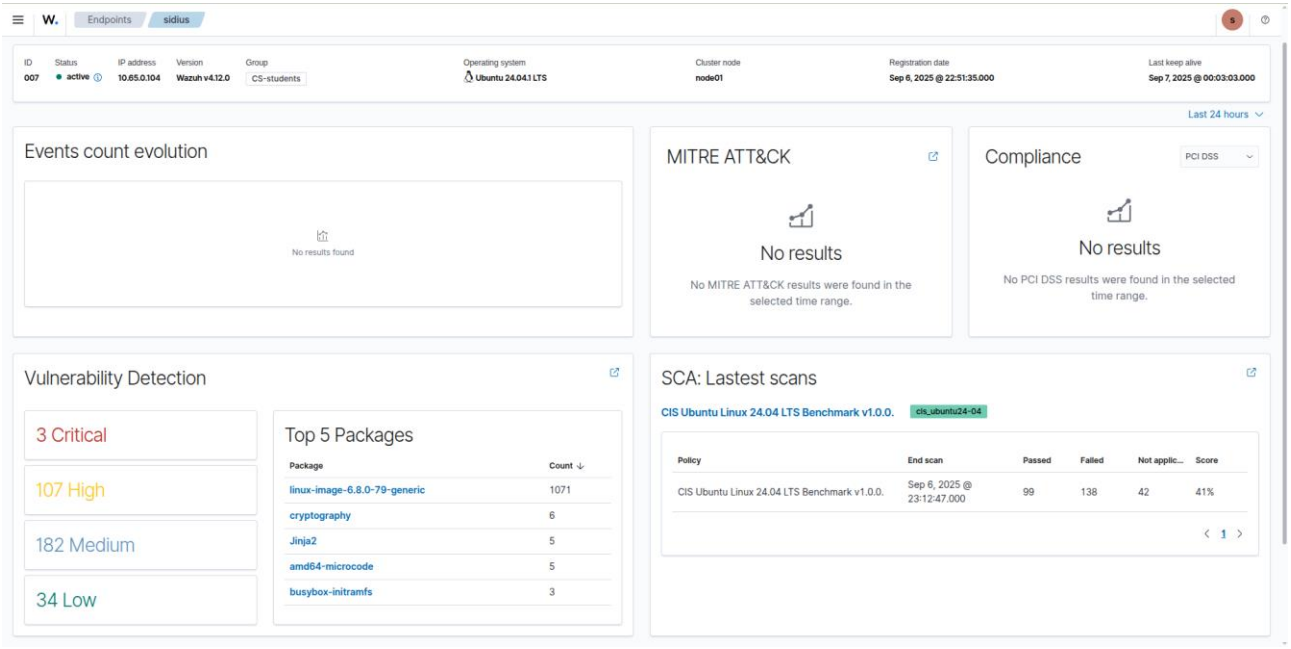
Vulnerability Detection – виявлення відомих вразливостей у середовищі.

MITRE ATT&CK – відображення подій відповідно до тактик і технік з бази MITRE ATT&CK для кращого розуміння атак.

Таким чином, ця панель дозволяє отримати огляд стану безпеки всієї інфраструктури: від активності агентів і подій різної критичності до аналізу конфігурацій, виявлення вразливостей та полювання на загрози.



Через пару хвилин хост з’явиться у панелі і через 10-15 хв сканування будуть показані всі виявленні вразливості даного хоста.



Необхідно дослідити графічний інтерфейс клієнтської машини після встановлення та підключення Wazuh-агента. Ознайомитися з можливостями веб-інтерфейсу Wazuh (Kibana), що відображає дані з клієнта:

- перевірити, чи з’явився новий агент у списку Agents Management;
- дослідити інформацію про клієнтську систему (версія ОС, статус підключення, IP-адреса);
- переглянути журнали подій, які надходять із клієнта;
- проаналізувати виявлені попередження та сигнали безпеки;
- побудувати просту візуалізацію (наприклад, кількість подій за певний період часу).

Контрольні питання

1. Яку інформацію відображає розділ Agents Summary і чому важливо стежити за статусом агентів?
2. Які рівні критичності подій виділяє Wazuh у блоці Last 24 Hours Alerts?
3. Для чого використовується модуль Configuration Assessment?
4. Які можливості надає функція File Integrity Monitoring?
5. Чим займається модуль Malware Detection і які типи загроз він допомагає виявити?
6. Які завдання вирішує інструмент Threat Hunting у Wazuh?
7. Як модуль Vulnerability Detection допомагає в управлінні інформаційною безпекою?
8. Що таке MITRE ATT&CK і чому інтеграція з ним є важливою для аналізу атак?
9. Які дії адміністратора можуть бути потрібні після виявлення критичних подій у панелі Wazuh?
10. Як Wazuh Dashboard допомагає в моніторингу клієнтських машин у реальному часі?

ЛАБОРАТОРНА РОБОТА №3

Тема: Оцінка ризиків та їхнє виправлення з використанням можливостей Wazuh на Ubuntu

Мета: Оцінити наявність і ризик вразливостей на клієнтській машині та виконати їхнє усунення з використанням можливостей Wazuh (виявлення вразливостей, інвентаризація, логування та верифікація виправлення)**.**

Необхідні інструменти:

1. **Операційна система**: Ubuntu (версія 20.04 або новіша).
2. **Wazuh agent –** клієнтський агент для збору логів та подій.
3. **Доступ до Wazuh Manager (сервера) –** попередньо налаштований сервер з Wazuh.
4. Права адміністратора (**sudo**) на клієнтській машині.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Рівні вразливостей (**Informational / Low / Medium / High / Critical**) у контексті Wazuh — приклади й короткі заходи пом'якшення.

Informational (Інформаційні)

Опис: не є вразливістю — це корисна телеметрія або конфіг-порада (наприклад, конфіг-файли без секретів, версія ПЗ, налаштування журналювання).

Приклад у Wazuh: агент надсилає детальні діагностичні логи або відомості про версії компонентів у відкритому доступі.

Ризик: низький — корисно для діагностики; але накопичення таких даних може полегшити розвідку для атак.

Мітки пом'якшення: обмежити доступ до діагностичної інформації, мінімізувати логування в продакшені, обмежити доступ до API/консолі.

Low (Низький)

Опис: невеликі проблеми, які навряд призведуть до компрометації; можуть створювати дрібні незручності або додаткові шумові сигнали.

Приклад у Wazuh: помилково налаштований рівень логів, незначні інформаційні витоки в логах, слабкі повідомлення у UI, дрібні недоліки в валідації полів форми.

Ризик: локальний/мінімальний вплив на конфіденційність або надійність.

Мітки пом'якшення: виправити валідацію вводу, мінімізувати чутливі поля у логах, застосувати принцип найменших привілеїв для ролей.

Medium (Середній)

Опис: дає можливість збору інформації або часткової ескалації прав; сам по собі часто не призводить до повного зламу, але полегшує подальші атаки.

Приклад у Wazuh: витік конфігураційних даних, недостатні обмеження Web UI, застарілі бібліотеки зі середнім CVE.

Мітки пом'якшення: ротація ключів, оновлення залежностей, перевірка ролей/ACL, шифрування каналів.

High (Високий)

Опис: серйозні вразливості, які можуть дозволити локальне підвищення привілеїв, несанкціонований доступ до значної кількості даних або порушити важливі функції.

Приклад у Wazuh: уразливість у REST API, що дозволяє змінювати конфіг агента; неправильно захищені збережені креденшали; SQL/NoSQL ін'єкції в модулі зберігання.

Ризик: значний — підвищує шанс отримати контроль над частиною інфраструктури або витягнути важливі дані.

Мітки пом'якшення: термінове патчення та розгортання, аудит доступів, шифрування секретів у сховищі (vault), застосування WAF/мережевих сегментацій.

Critical (Критичний)

Опис: найвищий пріоритет — дає можливість віддаленого виконання коду (RCE), повного обходу автентифікації або втрати цілісності/конфіденційності всієї системи. Потребує негайного реагування.

Приклад у Wazuh: віддалений RCE через компонент менеджера/агента; уразливість у механізмі аутентифікації, яка дозволяє автентифікуватися без пароля; витік master key.

Ризик: критичний — компрометація всієї SIEM/EDR, можливість lateral movement по мережі.

Мітки пом’якшення: негайний інцидент-респонс (izolate, take down), застосування екстрених патчів, ротація всіх ключів/сертифікатів, повний аудит та відновлення з надійних резервних копій.

Рекомендації для реагування по рівнях

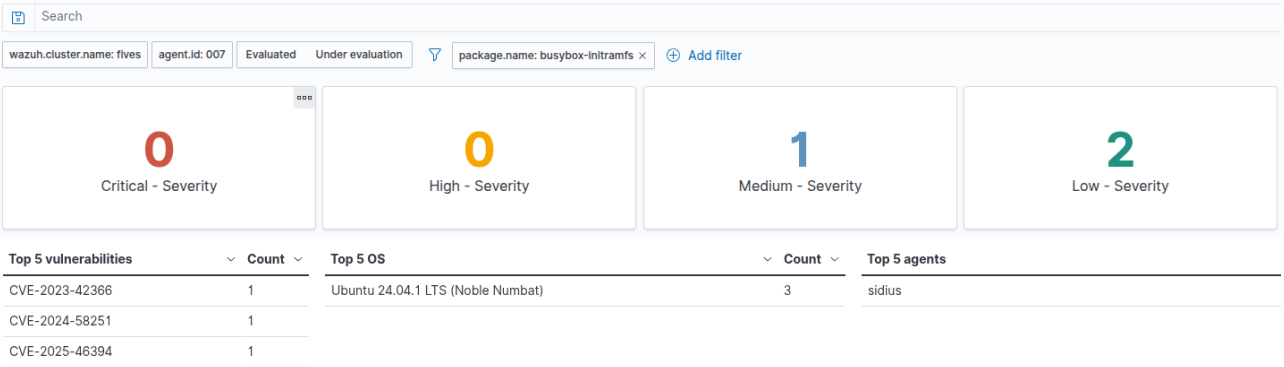
Informational / Low: запланувати виправлення в звичайному циклі релізів; налаштувати логування та мінімізувати «шум».

Medium: пріоритетна перевірка, тестування; ротація токенів/ключів; оновлення залежностей.

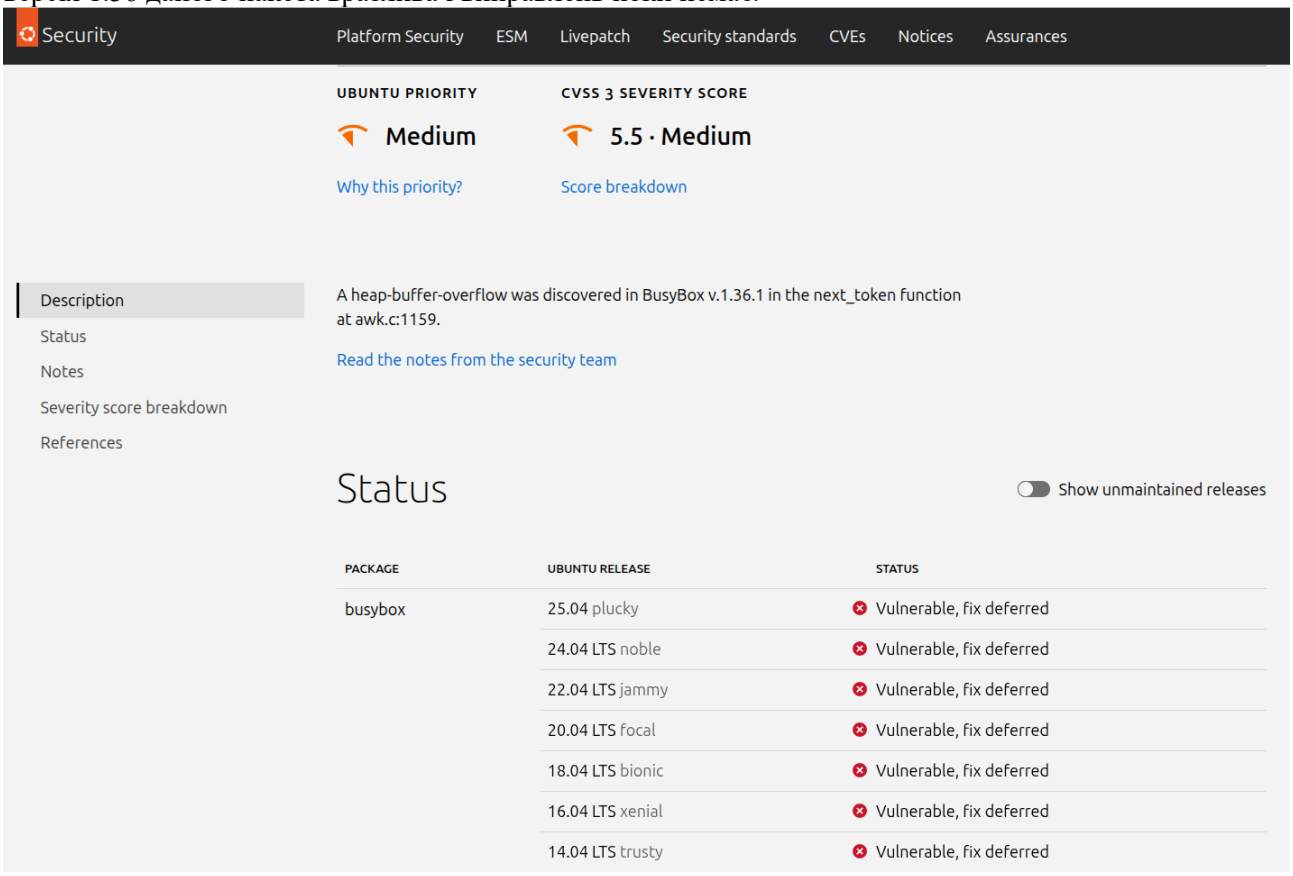
High: термінове виправлення — 24–72 години; аудит доступів; тимчасові обмеження сервісів.

Critical: негайний інцидент-респонс — ізоляція, патч, ротація секретів, зміна конфігурації, повний forensic.

Для перших виправлень даної машини візьмемо декілька вразливостей рівня Medium яких на даний момент на машині є 182, а саме вразливість пакету busybox-initramfs



Спочатку пошукаємо більше інформації про дані CVE вразливості з чого ми дізнаємося що версія 1.36 даного пакета вразлива і виправлень поки немає.



Завдання

- Ідентифікувати пов’язані CVE та оцінити критичність (CVSS, рівень у Wazuh).
- Зафіксувати ідентифікатори CVE, рівень ризику та рекомендації з джерел (наприклад, NVD).

- Виконати виправлення (remediation) на клієнті — оновлення/патч обраного пакета.

Оновлення пакетної бази і оновлення пакета:

sudo apt update

sudo apt install --only-upgrade busybox-initramfs

Якщо пакет відсутній у репозиторії або потрібен спеціальний патч — задокументувати процедуру отримання та встановлення оновлення (довідка дистрибутива, backport або вручну .deb).

Перевірити вплив виправлення та відновити роботу системи при потребі.

Повторна перевірка версії:

dpkg -s busybox-initramfs

Якщо оновлення вимагає перезавантаження initramfs/системи — виконати відповідні команди:

sudo update-initramfs -u

sudo reboot

Верифікувати усунення вразливості через Wazuh.

Дочекатися, поки агент знову надішле інвентаризацію (або вручну ініціювати оновлення інвентаря/повідомлення).

Перевірити у Wazuh Dashboard / Vulnerability Detection, що для busybox-initramfs більше немає відкритих CVE або що статус змінено на «mitigated»/«resolved».

Якщо Wazuh показує старі дані — переглянути логи агента на предмет помилок у передачі даних.

Задокументувати процедуру, результати та план відкату.

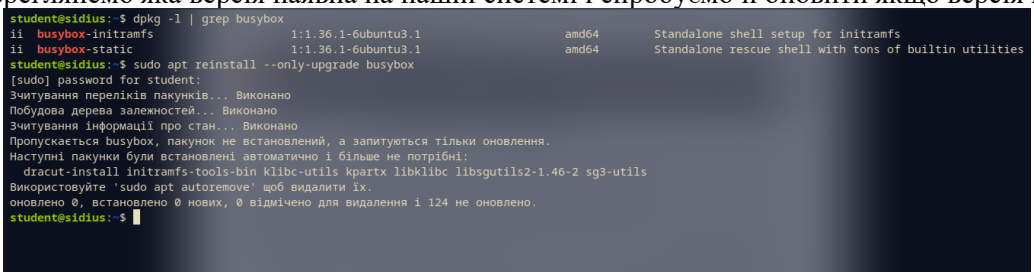
Описати застосовані команди, версії до і після, виявлені CVE, час виконання і результати перевірок.

Вказати план відкату (як повернутися до попереднього стану), тестову процедуру після оновлення та рекомендації щодо моніторингу.

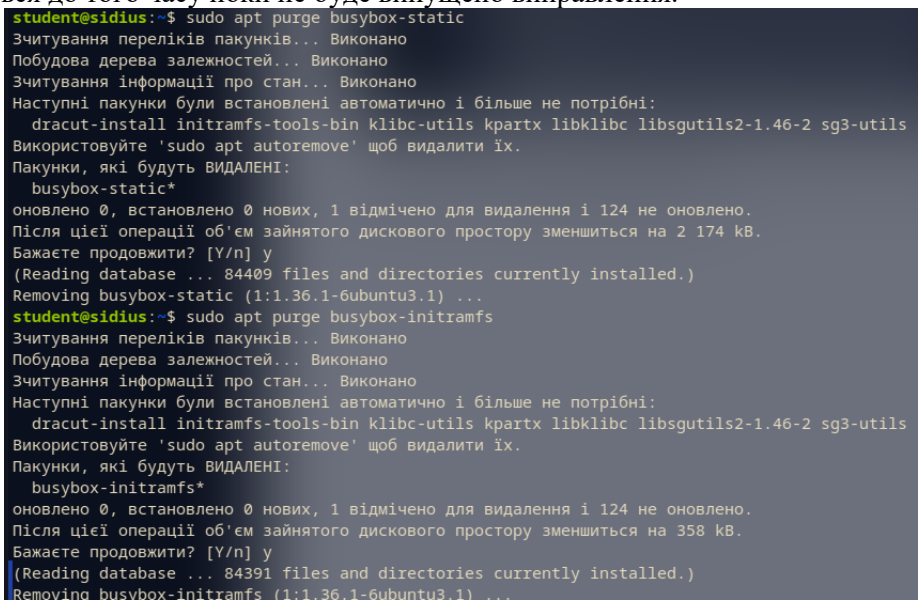
За потреби налаштувати додаткові правила/повідомлення у Wazuh.

Створити правило оповіщення для повторного виявлення критичних версій busybox-initramfs або налаштувати автоматичне створення інциденту/тікету.

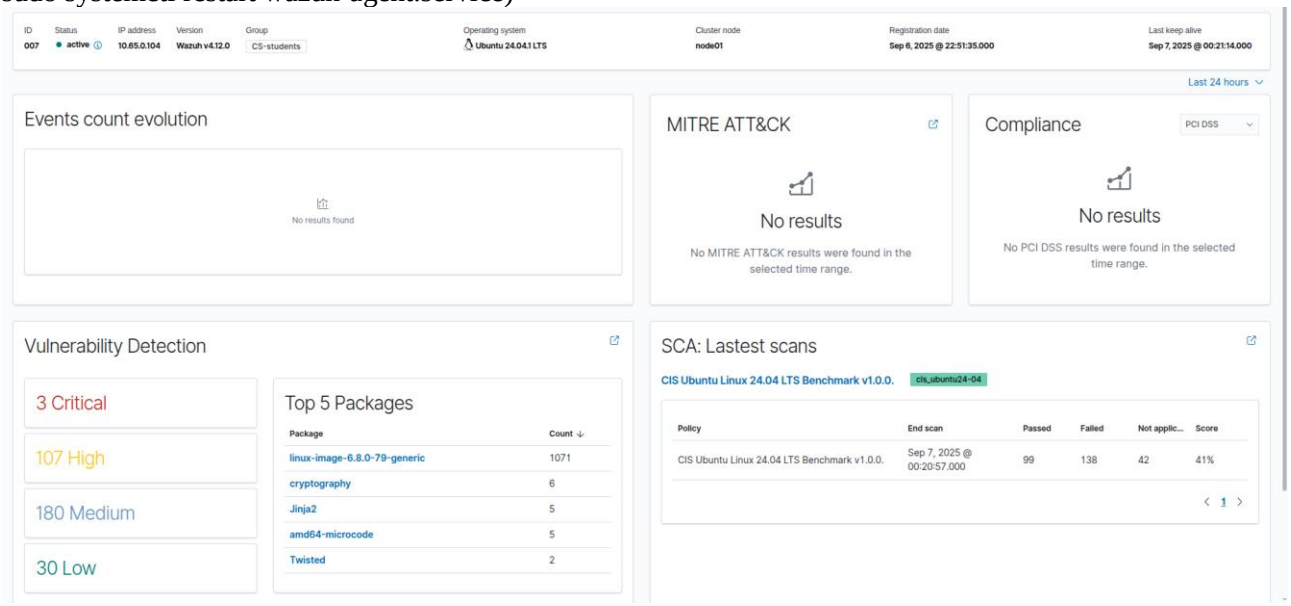
Переглянемо яка версія наявна на нашій системі і спробуємо її оновити якщо версія вразлива



Оскільки ні в системі ні на офіційному сайті не вказано про оновлення даного пакету, єдиний спосіб це повне видалення цього пакету, якщо він не критичний для системи і активно не використовується до того часу поки не буде випущено виправлення.



І через деякий час панель сканування видасть що дана вразливість зникла, а разом з не і кілька CVE вразливостей (потрібно перезапустити агента для миттєвого оновлення даних через команду - sudo systemctl restart wazuh-agent.service)



Контрольні питання

1. Що таке вразливість типу outdated package?
2. Чому використання застарілих бібліотек та пакетів становить загрозу безпеці?
3. Які рівні критичності (Informational, Low, Medium, High, Critical) може мати ця вразливість?
4. Який модуль Wazuh відповідає за перевірку пакетів на вразливості?
5. Як саме Wazuh отримує інформацію про CVE, пов’язані з встановленими пакетами?
6. Де в Wazuh можна переглянути сповіщення про застарілі пакети?
7. Яку команду потрібно виконати на Linux-системі, щоб вручну перевірити список застарілих пакетів?
8. Як у Wazuh можна налаштувати оповіщення, щоб воно відправляло повідомлення адміністратору про знайдену вразливість?
9. Які дані містить типове сповіщення Wazuh про застарілий пакет (назва, версія, CVE, рівень небезпеки тощо)?
10. Які основні кроки потрібно виконати для оновлення застарілого пакета?
11. Чим відрізняється часткове оновлення пакета від повного оновлення системи?
12. Чому важливо перевіряти сумісність після оновлення?
13. Як перевірити у Wazuh, що вразливість після оновлення усунена?
14. Які способи можна використати для створення звіту про оновлення пакетів у Wazuh?
15. Які ще дії (окрім оновлення) можна застосувати для зменшення ризику використання вразливих пакетів?

ЛАБОРАТОРНА РОБОТА №4

Тема: Оцінка ризиків та їхнє виправлення з використанням можливостей Wazuh на Windows
Мета: Оцінити наявність і ризик вразливостей, пов'язаних з пакетом на клієнтській машині та виконати їхнє усунення з використанням можливостей Wazuh (виявлення вразливостей, інвентаризація, логування та верифікація виправлення)**.**
Необхідні інструменти:

1. **Операційна система**: Windows 10.
2. **Wazuh agent –** клієнтський агент для збору логів та подій.
3. **Доступ до Wazuh Manager (сервера) –** попередньо налаштований сервер з Wazuh.
4. Права адміністратора (**sudo**) на клієнтській машині.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Wazuh Agent для Windows — по суті, це клієнтська частина Wazuh, яка встановлюється на хост і збирає дані для надсилання на Wazuh Manager.

На Windows агент ставиться як звичайний .msi інсталятор.

Після встановлення запускається як Windows service (OssecSvc).

Агент постійно виконує моніторинг системи та відправляє дані на менеджер. Основні модулі:

File Integrity Monitoring (FIM) — відстежує зміни у файлах/реєстрі (наприклад, system32, HKLM, HKCU).

Log Collection — збирає Windows Event Logs (Security, System, Application) і Sysmon-логи, якщо встановлено.

Software Inventory & Vulnerability Detection — знімає список встановлених пакетів, драйверів, бібліотек та порівнює з базою CVE.

Active Response — може виконати скрипт чи команду (наприклад, заблокувати IP через Windows Firewall) у відповідь на подію.

Rootcheck / Policy Monitoring — перевіряє систему на відомі шкідливі файли, небезпечні конфіги, політики безпеки.

Агент встановлює захищене TCP-з'єднання з Wazuh Manager (підтримується TLS).

Кожен агент має унікальний ключ (генерується при реєстрації).

Дані надсилаються у форматі JSON-подібних повідомлень.

На Windows-хості агент тільки збирає та пакує дані.

Вся логіка аналізу (правила, кореляція, виявлення атак) відбувається на Wazuh Manager.

Якщо спрацює правило, менеджер може ініціювати Active Response назад на агент (наприклад, kill процеса, блокування IP).

У Windows агента є власний лог-файл:

C:\Program Files (x86)\ossec-agent\ossec.log

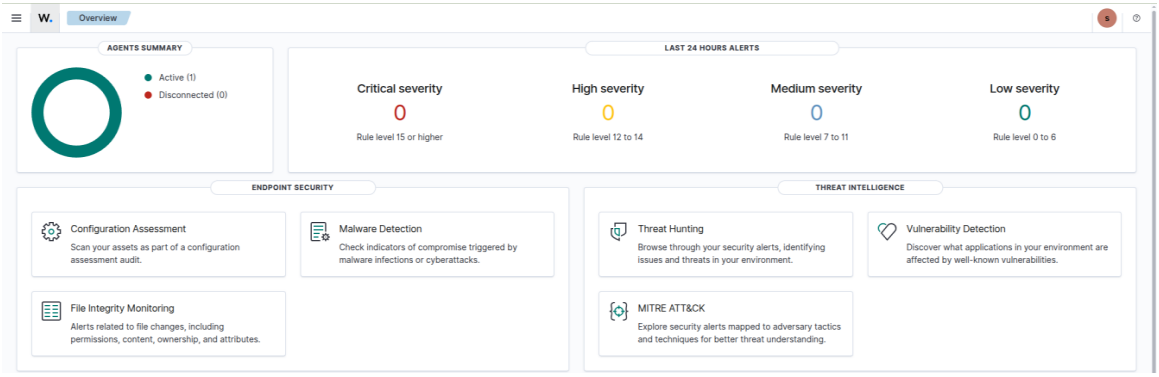
Там видно з'єднання з менеджером, помилки, статус відправки логів.

Якщо користувач запускає підозрілий .exe, Windows створює записи у Security Event Log та/або Sysmon. Wazuh Agent збирає ці події й відправляє їх на Manager. Manager застосовує правила кореляції визначає, що це підозріла активність. Якщо налаштовано, Manager віддає команду агенту агент запускає Active Response (наприклад, додає IP у Windows Firewall).

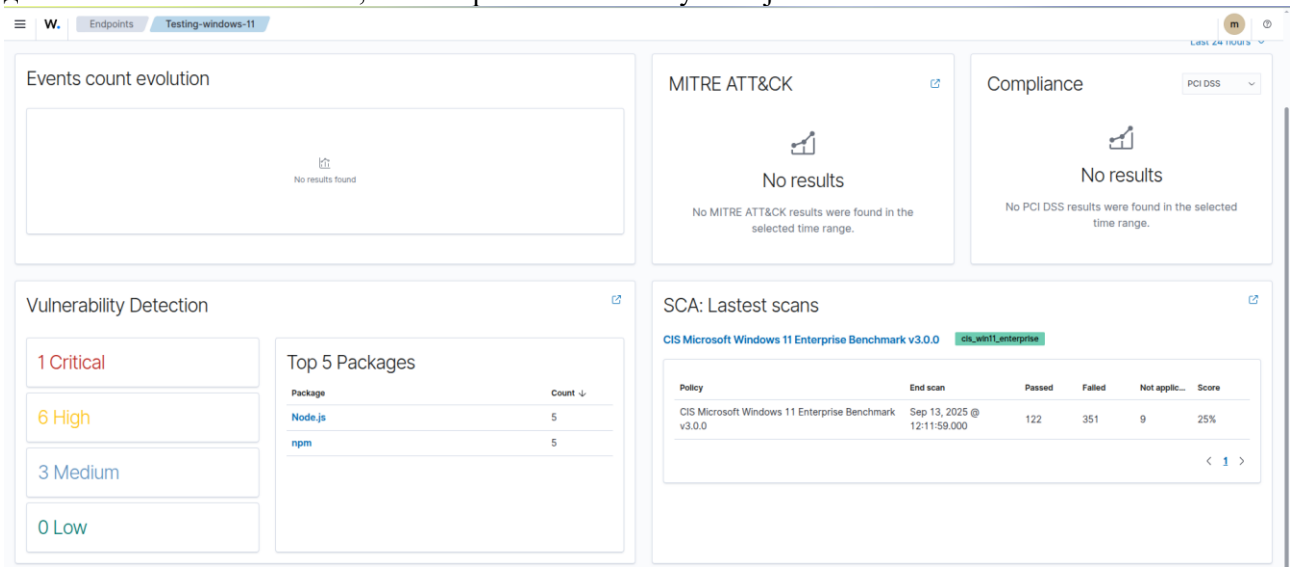
Завдання

- Ідентифікувати пов'язані CVE та оцінити критичність (CVSS, рівень у Wazuh) для ОС WINDOWS, виконати виправлення.
- Запустити на клієнтській машині виконуваний файл (https://drive.google.com/open?id=1H4R5y3Ml-ydyz2nouRxsVTxt1p19ng54&usp=drive_fs), що додає себе в автозавантаження та копіює та проаналізувати отримані логи та реакцію.

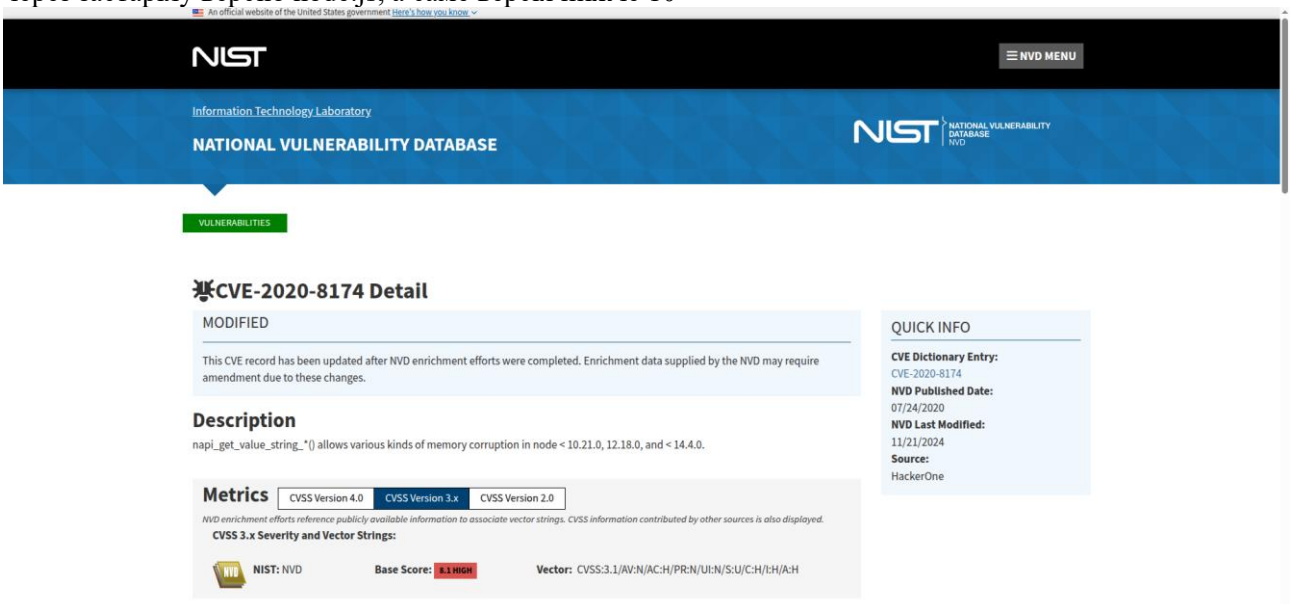
Для виконання завдання необхідно скористатись цифровою інфраструктурою навчальної лабораторії тільки у ЛОКАЛЬНІЙ мережі ЗУНУ за посиланням:
https://10.65.0.106:8888/ і увійти на сайт за виданими обліковими записами.



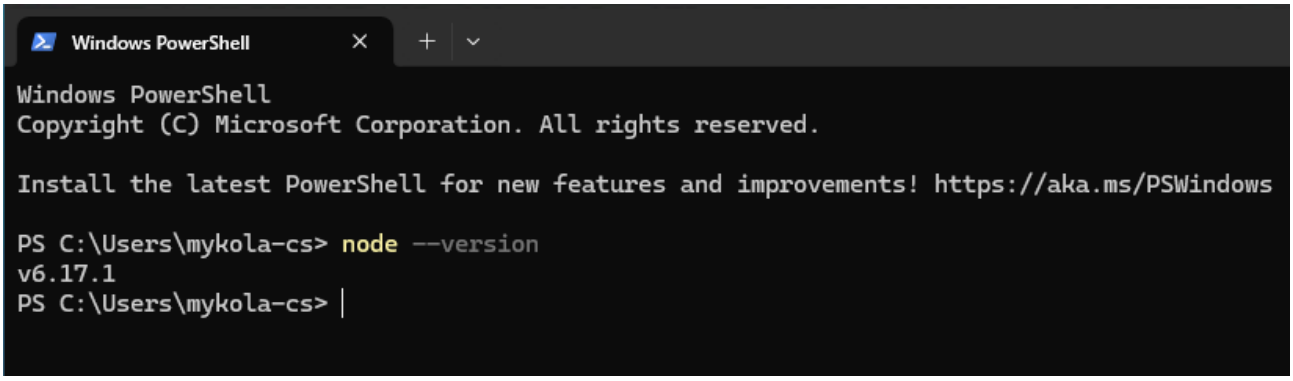
Для перших виправлень на ОС WIndows машині візьмемо вразливість рівня Critical яка на даний момент на машині є 1, а саме вразливість пакету node.js.



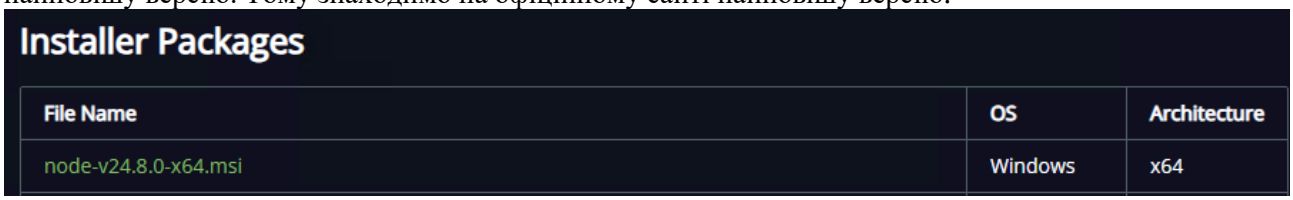
Спочатку пошукаємо більше інформації про дані CVE вразливості з чого ми дізнаємося що це через застарілу версію node.js, а саме версія нижче 10



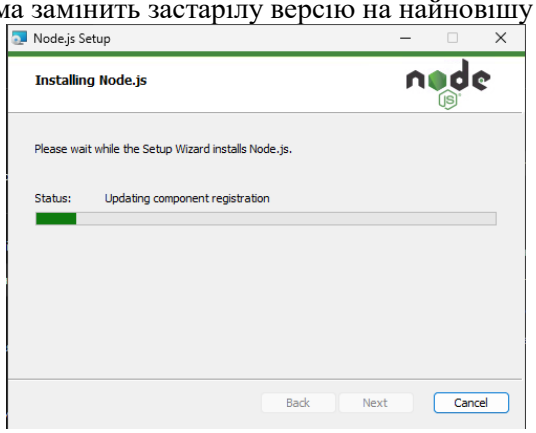
Переглянемо яка версія наявна на нашій системі і спробуємо її оновити якщо версія вразлива, зайшовши через RDP або SSH на встановленого вами агента або наданій вам машині.



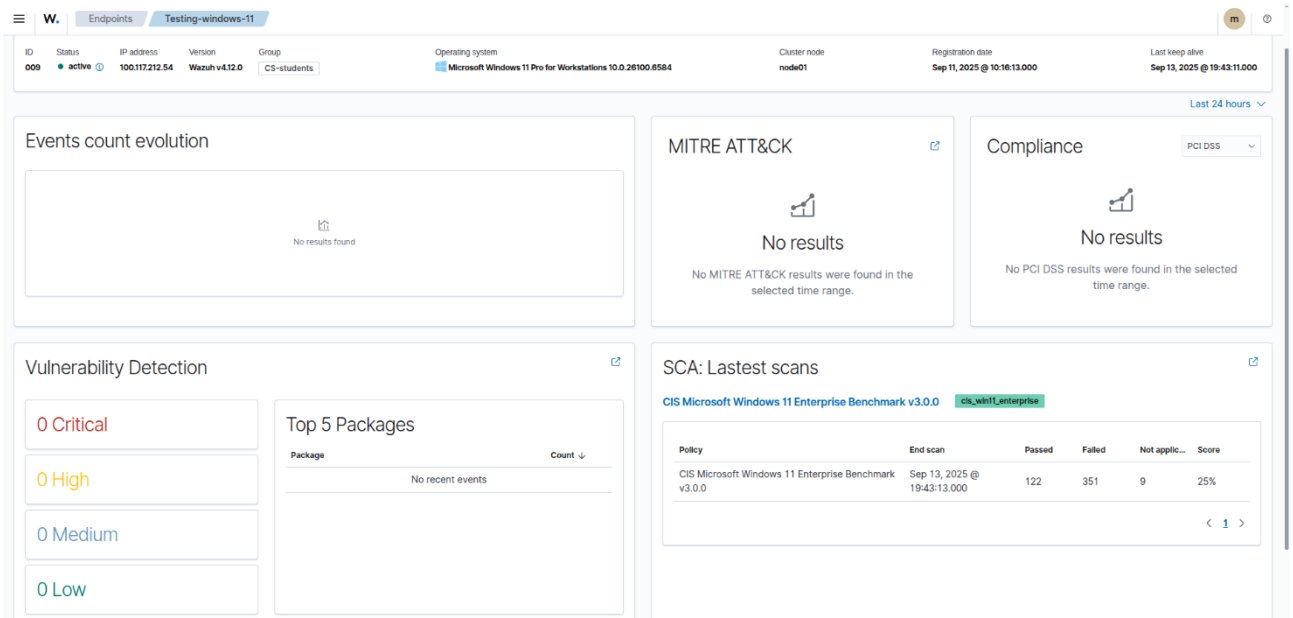
Тому для вирішення даної вразливості вистачить всього лиш заміни даного пакету на найновішу версію. Тому знаходимо на офіційному сайті найновішу версію.



При інсталяції програма замінить застарілу версію на найновішу



І через деякий час панель сканування видасть що дана вразливість зникла, а разом з не і кілька CVE вразливостей (потрібно перезапустити агента для миттєвого оновлення даних через команду в PowerShell від імені адміністратора - Restart-Service -Name wazuh)



Контрольні питання

1. Яке призначення Wazuh Agent у Windows?
2. У чому різниця між Wazuh Agent і Wazuh Manager?
3. Які основні модулі входять до складу агента на Windows?
4. Які журнали подій збирає Wazuh Agent за замовчуванням?

5. Яким чином Wazuh Agent встановлюється у Windows і в якому вигляді працює?
6. Як відбувається зв’язок між Wazuh Agent і Wazuh Manager?
7. Для чого агенту потрібен ключ реєстрації?
8. Де зберігаються лог-файли самого агента у Windows?
9. Як Wazuh Agent у Windows реалізує File Integrity Monitoring (FIM)?
10. Що таке Active Response і які дії він може виконувати на Windows-хості?
11. Як агент може відслідковувати зміни у Windows Registry?
12. Чим відрізняється збір подій з Windows Event Logs від збору Sysmon-логів?
13. Яку команду використовують для перевірки статусу служби агента у Windows?
14. Як перевірити, чи агент успішно підключився до Wazuh Manager?
15. Які кроки потрібно виконати, щоб зареєструвати новий агент у Wazuh Manager?
16. Як у Kibana / Wazuh Dashboard переглянути події, які надійшли від Windows-агента?
17. Чому важливо шифрувати канал між агентом та менеджером?
18. Які ризики виникають, якщо агент не відправляє логи або втрачає з’єднання з менеджером?
19. Як можна обмежити дії Active Response у Windows для уникнення помилкових блокувань?
20. Які переваги надає використання Sysmon у поєднанні з Wazuh Agent?
21. Автоматична перевірка оновлень пакетів і CVE.

ЛАБОРАТОРНА РОБОТА №5

Тема: Ознайомлення з інтерфейсом MISP та створення повідомлення про загрозу**.**

Мета: Ознайомитись з інтерфейсом MISP, набути навиків використання повідомлень про загрози та створення власних повідомлень в межах установи**.**

Необхідні інструменти:

1. **Доступ до MISP (сервера) –** попередньо налаштований сервер з Wazuh.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Основні можливості MISP:

Збір і зберігання подій (Events) - створення подій із комплектом IOC/атрибутів (IP, домен, хеші, URL, YARA, email тощо), описом інциденту і метаданими (дата, автор, теги).

Класисифікація й таксономії - теги, категорії, таксономії і «галактики» (galaxies) для структурованого опису APT-груп, malware-сімейств, TTP (techniques), кампаній.

Кореляція та зв’язування (Correlation engine) - автоматичне зв’язування і IOC-кореляція між подіями — допомагає виявляти повтори і поширення індикаторів.

Обмін і синхронізація - синхронізація з іншими MISP-серверами або організаціями (push/pull), налаштовувані sharing groups/organizations/permissions.

API для автоматизації - повнофункціональний REST API (JSON) для створення/отримання подій, атрибутів, пошуку індикаторів і інтеграції з SIEM/EDR/скриптами.

Фіди та інтеграція модулів (MISP Modules) - підтримка модулів для автоматичного збагачення (DNS/WHOIS, VirusTotal, PassiveTotal і т.д.) та імпорту/експорту з інших джерел.

SIGHTINGS (підтвердження спостережень)

Можливість фіксувати sightings — чи бачили ви цей IOC у вашій інфраструктурі (частота, дата, контекст).

Формати експорту/імпорту - підтримка JSON, CSV, STIX/STIX2, OpenIOC та ін. — зручно для обміну з зовнішніми системами.

Warninglists і списки блокувань - створення та управління чорними/білосписами для автоматичної фільтрації та експорту в мережеві пристрої/файрволи.

PGP-підпис і підтвердження джерела - можливість підписувати/верифікувати події та імпорти через PGP (на скрині видно посилання на «Server PGP public key»).

Ролі та контроль доступу - гнучкі права доступу (організації, ролі, ACL), можливість обмежити бачення/зміни даних.

Історія змін і пропозиції (proposals) - версіонування подій, механізм пропозицій змін і голосування/прийняття змін.

Типові сценарії використання - централізоване Threat Intelligence сховище для SOC і CSIRT.

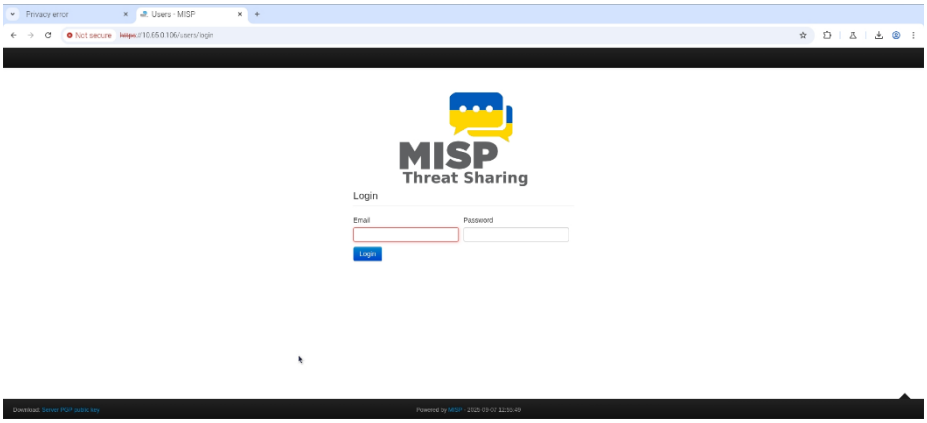
Автоматична розсилка IOC до SIEM/EDR (наприклад, Wazuh, Elastic) через API/скрипти.

Обмін розвідувальною інформацією з партнерами/асоціаціями - збагачення інцидентів (IOC enrichment) перед їхнім кореляційним аналізом.

Завдання:

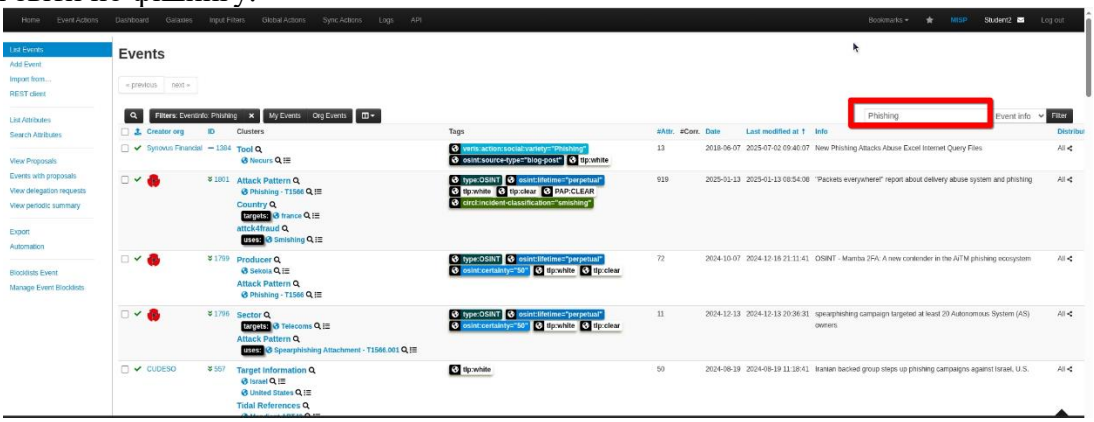
- Здійснити огляд можливостей MISP та поточних доступних повідомлень про загрози.
- Створити власне повідомлення про загрозу в межах установи.

Для виконання завдання необхідно скористатись цифровою інфраструктурою навчальної лабораторії кафедри тільки у ЛОКАЛЬНІЙ мережі ЗУНУ, за посиланням: https://10.65.0.106/ і увійти на сайт за виданими обліковими записами.



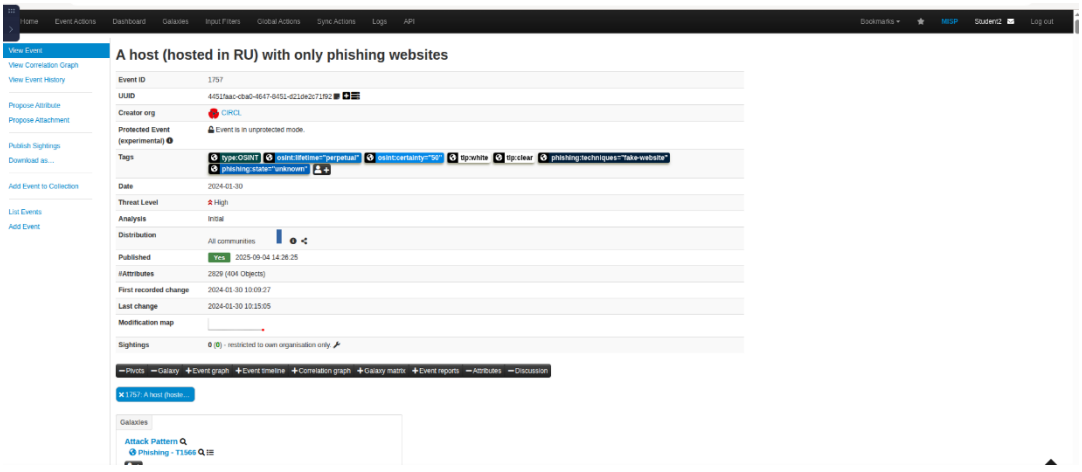
Ознайомлення з інтерфейсом MISP

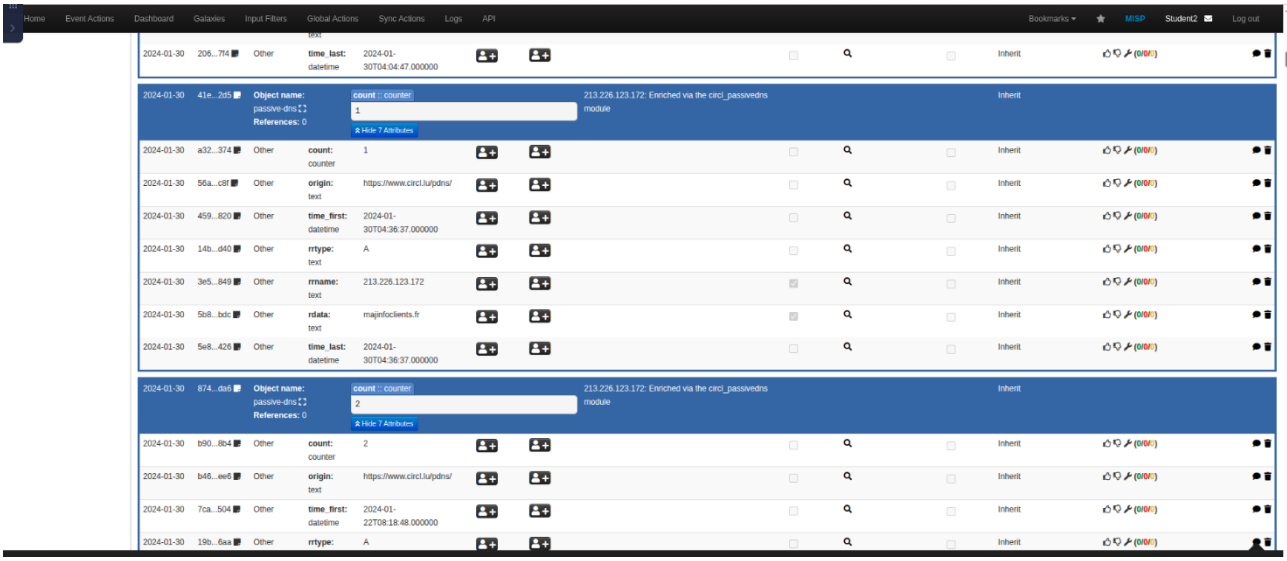
Для початку перейдемо на вкладку Event Actions де будуть показані всі останні звіти по атакам і вразливостям від різних організацій по всьому світу, для прикладу переглянемо останні звіти по фішингу.



За потреби ми можемо сортувати їх для зручності і візьмемо один з них для більш детального огляду клікнувши по ID даної події.

На сторінці події можемо ознайомитись з її описом, відрізки часу її дії, а нижче у списку можна побачити структуровані об’єкти які створюються при перевірці даної події іншими система і подають автоматичний звіт з доменами, хешами і адресами для автоматичного виявлення цієї атаки на інших системах.

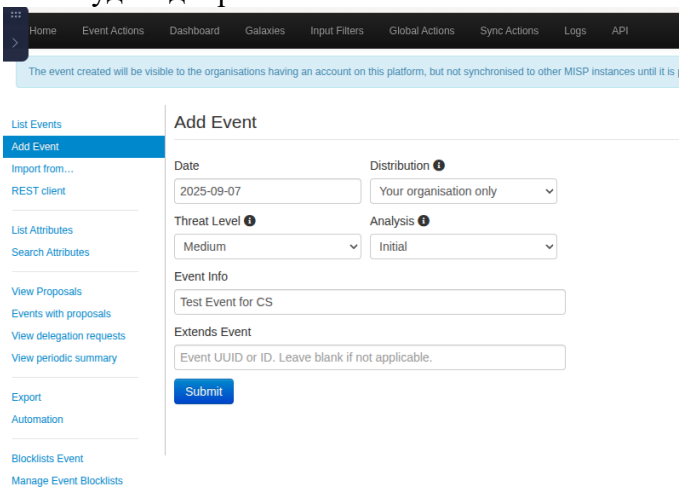




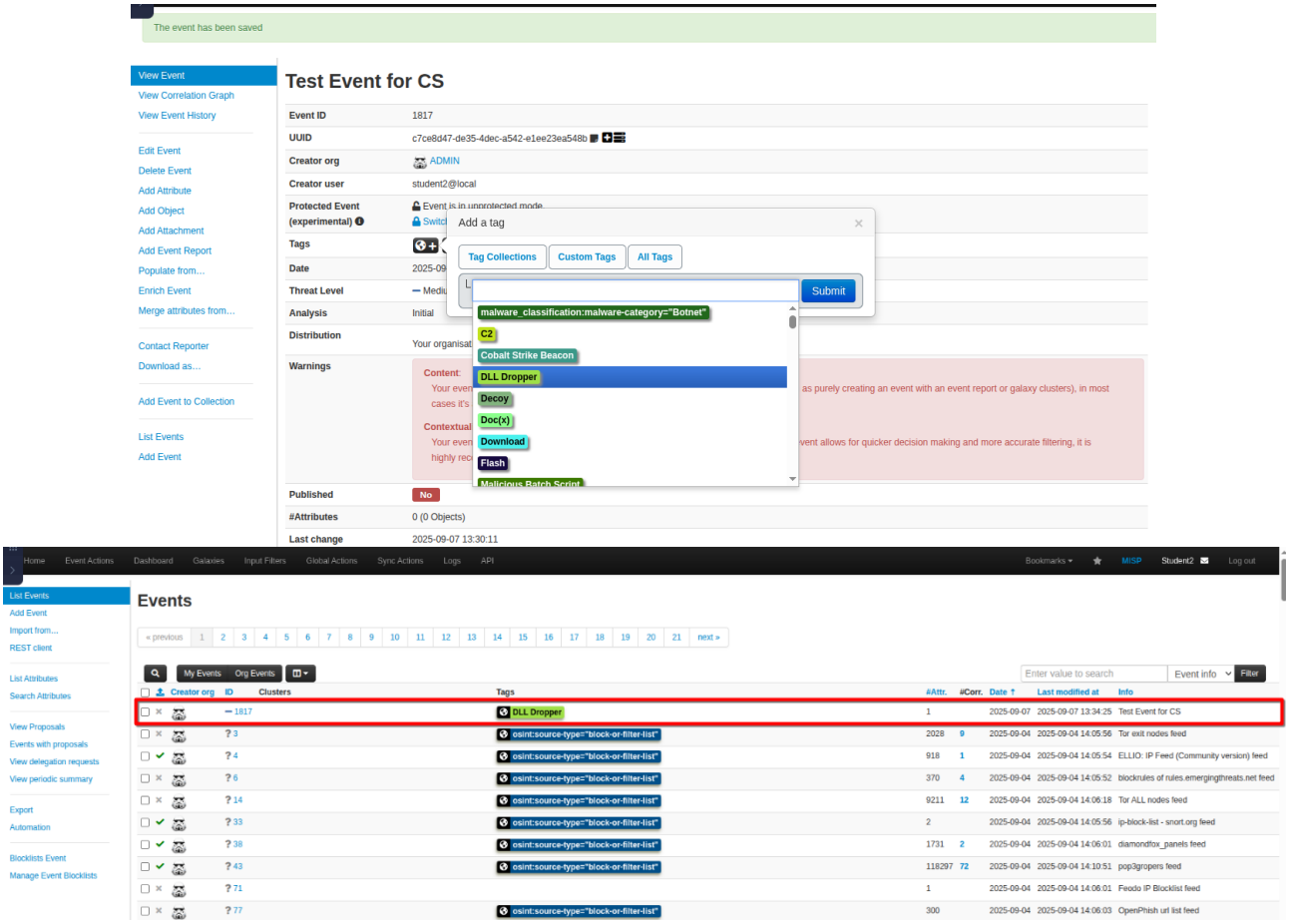
Тому основним поки завданням буде оглянути кілька таких подій і коротко розписати які дії вони виконують, які домени і адреси використовуються і можливі контрзаходи проти них.

Створення власної події загрози

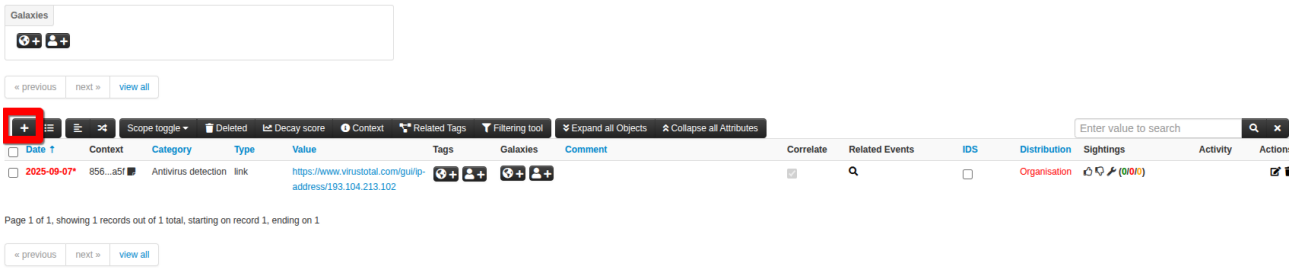
У разі виявлень власних вразливостей чи атак, нам потрібно створити власну подію в якій опишемо основні атрибути і властивості даної події, рівень загрози і короткий опис. Поки створимо тестову події яка буде відображатись тільки нашій локальній організації.

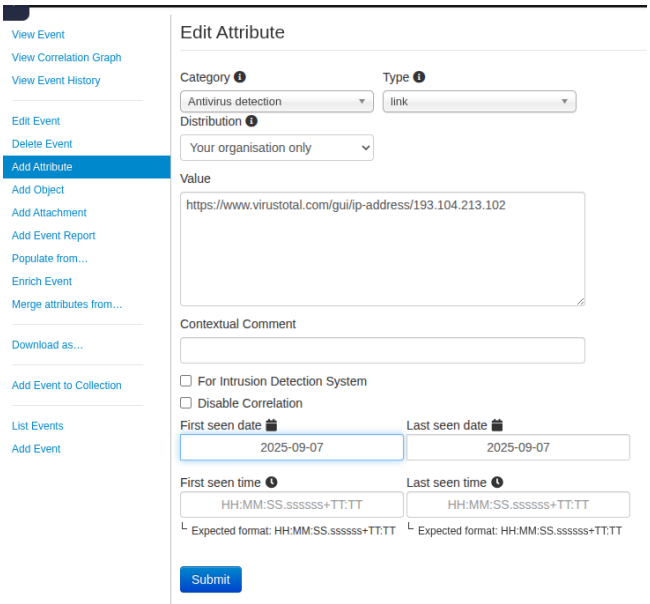


Далі ми можемо додати теги до події щоб легше її ідентифікувати і поєднювати з іншими схожими, для прикладу виберемо що наша подія це DLL Dropper



Далі ми можемо додати свої об’єкти для детальнішого огляду даної події і вказати на типові ознаки для даної атаки щоб протидіяти їй у майбутньому, для прикладу сканування на VirusTotal





Тепер повернувшись на сторінку подій ми можемо побачити нашу недавно створену подію загрози, яка поки що висвітлюється лиш нашій локальній організації, але її також можна поширити і інших по всьому світу, якщо вона буде представляти реальну загрозу, щоб при її

виявлені у інших системах, вона зразу автоматично блокувалась по створених нами сигнатурних об’єктах

Контрольні питання

2. Що таке MISP і для чого він використовується?
3. Які основні типи даних можна зберігати в MISP?
4. Що таке IOC (Indicator of Compromise) і як вони представлені в MISP?
5. Які переваги централізованого обміну Threat Intelligence?
6. Які основні компоненти входять до MISP (події, атрибути, теги, таксономії)?
7. Що таке «галактики» (galaxies) у MISP?
8. Для чого використовується PGP у MISP?
9. Які формати імпорту та експорту підтримує MISP?
10. Як створюється нова подія в MISP?
11. Які основні категорії атрибутів можна додавати до події?
12. Що таке sightings і як вони використовуються?
13. Чим корисні warninglists у роботі з IOC?
14. Як відбувається синхронізація між двома інстансами MISP?
15. Що таке sharing groups і яку роль вони відіграють?
16. Як можна інтегрувати MISP із зовнішніми системами (SIEM, EDR)?
17. Які можливості надає API MISP?
18. Які ролі користувачів існують у MISP і чим вони відрізняються?
19. Як захистити веб-інтерфейс MISP (TLS, автентифікація, права доступу)?
20. Чому важливо робити резервні копії бази MISP?
21. Які ризики можуть виникнути при неправильному налаштуванні доступу до MISP?

ЛАБОРАТОРНА РОБОТА №6

Тема: Системи виявлення та запобігання вторгнень (IDS/IPS)

Мета роботи: Ознайомитися з принципами роботи систем виявлення та запобігання вторгнень (IDS/IPS), навчитися налаштовувати та аналізувати їх роботу для підвищення рівня безпеки мережі.

Необхідне програмне забезпечення:

- **Snort** – система виявлення вторгнень з відкритим кодом. Вона фіксує загрози на основі визначених правил (правила для виявлення атак) та генерує оповіщення.
- **Suricata** – високопродуктивна система IDS/IPS, яка підтримує роботу як в режимі виявлення (IDS), так і в режимі запобігання вторгнень (IPS). Вона також підтримує високу пропускну здатність і аналізує не лише трафік, а й застосовує правила блокування атак.

Завдання:

- Встановлення та базове налаштування Snort.
- Встановлення та базове налаштування Suricata
- Використати Snort та Suricata John the Ripper для запобігання вторгнень.
- Аналіз результатів

ТЕОРЕТИЧНІ ВІДОМОСТІ:

IDS (Intrusion Detection System) – система виявлення вторгнень. Вона здатна виявляти підозрілі події в мережі, аналізуючи трафік і генеруючи сигнали для адміністратора.

- Основне завдання IDS – спостереження за мережевим трафіком, виявлення атак і інформування про них.

IPS (Intrusion Prevention System) – система запобігання вторгненням. Це вдосконалена версія IDS, яка не лише виявляє підозрілу активність, але й активно блокує її.

- IPS може блокувати шкідливий трафік в реальному часі, що робить її більш потужною у порівнянні з IDS.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

Завдання 1: Встановлення та базове налаштування Snort

1. **Встановлення Snort:** Щоб почати роботу з Snort, його необхідно встановити на віртуальну машину або фізичний сервер. У Kali Linux можна використовувати пакетний менеджер, наприклад:

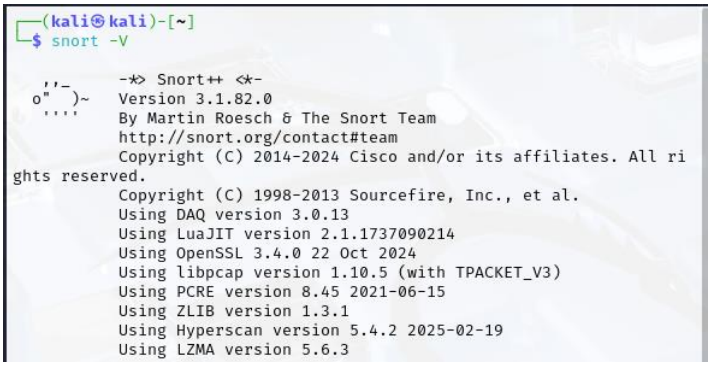
   sudo apt update && sudo apt install snort

   Якщо ви використовуєте інший дистрибутив, метод установки може змінюватися.

Перевірка встановлення Snort: Після успішної інсталяції, для перевірки версії і правильності установки можна використати команду:

snort -V

Команда виведе інформацію про версію Snort і підтвердить, що програма встановлена та готова до роботи.



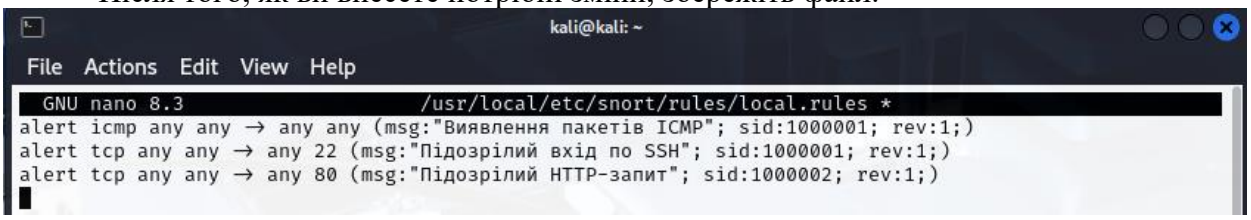
Step-by-Step Guide to Install and Configure Snort3 on Kali Linux: https://thesecmaster.com/blog/step-by-step-guide-to-install-and-configure-snort3-on-kali-linux-with-auto-rule-u

2. **Налаштування базових правил для виявлення підозрілих пакетів.** Snort використовує конфігураційні файли, в яких задаються правила для виявлення атак і небажаних

мережевих активностей. Для налаштування цих правил необхідно редагувати відповідний файл, наприклад, /usr/local/etc/snort/rules/local.rules, додаючи необхідні записи:

sudo nano /шлях/до/файлу/local.rules

Після того, як ви внесете потрібні зміни, збережіть файл.



Правила визначають, які пакети будуть вважатися підозрілими, а також в яких випадках система генеруватиме сповіщення про можливу атаку чи іншу підозрілу активність. Приклади налаштування базового правила:

Виявлення пакетів ICMP (ping-запити):

alert icmp any any -> any any (msg:"Виявлення пакетів ICMP"; sid:1000001; rev:1;)

- alert - створення сповіщення при виявленні пакета, що відповідає правилу.
- icmp - протокол, для якого правило спрацьовує (ICMP - протокол для команд ping).
- any any - будь-яка IP-адреса і будь-який порт.
- msg:"text" - повідомлення та його текст, що відобразиться в журналі при спрацьовуванні правила.
- sid:1000001- унікальний ідентифікатор правила.
- rev:1 - номер версії правила, якщо правило оновлюється або модифікується, номер ревізії збільшують (rev:2; rev:3; і т.д.)

Виявлення підозрілих пакетів TCP-трафіку для з'єднань через SSH:

alert tcp any any -> any 22 (msg:"Підозрілий вхід по SSH"; sid:1000001; rev:1;)

- alert tcp - активує сповіщення для TCP-трафіку.
- any 22 - спрацьовує для будь-якого джерела, яке підключається до порту 22.

Виявлення підозрілих HTTP-запитів, що можуть свідчити про сканування або інші атаки через веб-сервер:

alert tcp any any -> any 80 (msg:"Підозрілий HTTP-запит"; sid:1000002; rev:1;)

- tcp - використовуваний протокол (TCP для HTTP).
- any 80 - будь-яка IP-адреса на порт 80 (стандартний порт для HTTP).

Виявлення DNS-запитів або аномальних звернень до DNS-серверів:

alert udp any any -> any 53 (msg:"DNS-запит виявлено"; sid:1000003; rev:1;)

- udp - використовується протокол UDP, який застосовується для DNS.
- any 53 - будь-яка IP-адреса на порт 53 (стандартний порт для DNS).

Виявлення спроб підключення до нестандартного порту на цільовій машині:

alert tcp any any -> any 6660:6669 (msg:"Підозрілі спроби підключення до IRC-портів"; sid:1000004; rev:1;)

- any 6660:6669 - будь-яка IP-адреса на діапазон портів IRC (6660-6669).

Виявлення підозрілих FTP-запитів:

alert tcp any any -> any 21 (msg:"Підозрілі запити до FTP-сервера"; sid:1000005; rev:1;)

- any 21 - будь-яка IP-адреса на порт 21 (стандартний порт для FTP).

Виявлення великої кількості SYN-пакетів, що може вказувати на атаку типу SYN-флуд:

alert tcp any any -> any any (flags:S; msg:"Можлива атака SYN-флуд"; sid:1000006; rev:1;)

- flags:S - флаг SYN в TCP-пакеті, що означає початок з'єднання.
- any any - будь-яка IP-адреса та будь-який порт джерела.
- any any - будь-яка IP-адреса та будь-який порт призначення.

Перевірка файлу конфігурації Snort (починаючи з Snort 3):

sudo snort -T -c /шлях/до/файлу/snort.lua

Якщо немає помилок, Snort готовий до роботи.

```
Finished /etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
--------------------------------------------------
ips policies rule stats
              id  loaded  shared enabled    file
               0     208       0     208    /etc/snort/snort.lua
--------------------------------------------------
rule counts
       total rules loaded: 208
                text rules: 208
             option chains: 208
             chain headers: 1
--------------------------------------------------
service rule counts          to-srv  to-cli
                   file_id:      208     208
                     total:      208     208
--------------------------------------------------
fast pattern groups
                 to_server: 1
                 to_client: 1
--------------------------------------------------
search engine (ac_bnfa)
appid: MaxRss diff: 2236
appid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to passive.

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

3. **Запуск Snort у режимі виявлення вторгнень:** Запустіть Snort у реальному часі для моніторингу трафіку.

sudo /шлях/до/папки/snort -c /шлях/до/файлу/snort.lua -R /шлях/до/файлу/local.rules

Якщо служба працює, повинні відображатися налаштовані правила

```
--------------------------------------------------
ips policies rule stats
              id  loaded  shared enabled    file
               0     210       1     210    /usr/local/etc/snort/snort.lua
--------------------------------------------------
rule counts
       total rules loaded: 210
          duplicate rules: 1
                text rules: 210
             option chains: 210
             chain headers: 4
--------------------------------------------------
port rule counts
             tcp     udp    icmp      ip
     any       0       0       1       0
     dst       1       0       0       0
   total       1       0       1       0
--------------------------------------------------
```

Якщо конфігурація повинно відображатися повідомлення:

```
Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

4. **Тестове сканування мережі.** Для цього потрібно призначити свій мережевий інтерфейс для прослуховування трафіку з ключем -i та виводити повідомлення попередження на консолі з ключем -A. Наприклад:

sudo /шлях/до/папки/snort -c /шлях/до/файлу/snort.lua -R /шлях/до/файлу/local.rules -i eth0 -A alert_fast

sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -R /usr/local/etc/snort/rules/local.rules -i eth0 -A alert_fast

Щоб перевірити правильність роботи Snort, можна виконати тестування мережевого з’єднання за допомогою:

ping <IP-адреса системи>

```
Commencing packet processing
++ [0] eth0
02/19-08:20:41.415001 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::4ccb
:4db4:41c0:a3f2 → ff02::2
02/19-08:26:04.618667 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::dc46
:120:a257:e680 → ff02::16
02/19-08:26:04.644098 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::dc46
:120:a257:e680 → ff02::16
02/19-08:26:04.660241 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::dc46
:120:a257:e680 → ff02::16
02/19-08:26:04.660643 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::dc46
:120:a257:e680 → ff02::16
02/19-08:26:05.009247 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} fe80::dc46
:120:a257:e680 → ff02::16
```

Це підтверджує, що Snort працює належним чином.

Або здійснити тестове сканування мережі за допомогою nmap. Наприклад:

nmap -sP <IP-адреса системи>

Це дозволить перевірити мережу на наявність живих хостів.

Налаштування кількох файлів правил Snort

Якщо необхідно використовувати один самий файл правил кілька разів або декілька правил із різних файлів для запуску Snort можнае визначити файл кількох правил за різними шляхами у файлі конфігурації snort.lua. Для цього необхідно відредагувати snort.lua за допомогою nano або будь-якого текстового редактора CLI.

sudo nano /usr/local/etc/snort/snort.lua

Оновіть конфігурації, позначені червоним

```
-- 1. configure defaults

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = 'any'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = 'any'

include 'snort_defaults.lua'
```

Розкоментуйте рядок enable_builtin_rules і встановіть для нього значення true. Додайте шлях до файлу правил для кожного у новому рядку з директивою, include = <шлях/до/файлу/правил>:

```
-- 5. configure detection

references = default_references
classifications = default_classifications

ips =
{
    -- use this to enable decoder and inspector alerts
    enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)
    include = "/usr/local/etc/snort/rules/local.rules",
    variables = default_variables
}
```

В результаті система працюватиме за розширеним набором правил:

```
02/19-15:48:45.794702 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:45.806828 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:45.903634 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:46.159604 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} 35.214.136.
02/19-15:48:46.189693 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:46.602702 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:46.602702 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3]
02/19-15:48:47.360683 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} 35.214.136.
02/19-15:48:47.836867 [**] [116:414:1] "(ipv4) IPv4 packet to broadcast dest address" [**] [Priority: 3
02/19-15:48:49.255947 [**] [1:1000001:1] "Виявлення пакетів ICMP" [**] [Priority: 0] {ICMP} 35.214.136.
02/19-15:48:52.854567 [**] [116:414:1] "(ipv4) IPv4 packet to broadcast dest address" [**] [Priority: 3
02/19-15:48:57.872363 [**] [116:414:1] "(ipv4) IPv4 packet to broadcast dest address" [**] [Priority: 3
```

Завдання 2: Використання Suricata для запобігання вторгнень

1. **Встановлення Suricata та запуск в режимі IPS:** Для встановлення **Suricata** виконайте:

sudo apt-get install suricata

Запустіть її в режимі IPS для реального блокування шкідливого трафіку:

suricata -c /etc/suricata/suricata.yaml -i eth0

- **-c /etc/suricata/suricata.yaml**: вказує на конфігураційний файл.
- **-i eth0**: вказує на мережевий інтерфейс для моніторингу.

2. **Додавання правила для блокування підозрілих HTTP-запитів:** Для блокування HTTP-трафіку можна додати таке правило:

drop tcp any any -> any 80 (msg:»Suspicious HTTP traffic blocked»; sid:1000002; rev:1;)

- **drop**: означає, що якщо пакет відповідає правилу, він буде заблокований.
- **tcp any any -> any 80**: для TCP трафіку, який приходить на порт 80 (HTTP).
- **msg**: повідомлення про блокування.

3. **Перевірка ефективності правил:** Після налаштування правил можна виконати HTTP-запити через браузер або за допомогою curl, щоб перевірити, чи дійсно блокуються підозрілі запити:

curl http://example.com

Також можна здійснити сканування за допомогою **nmap**, щоб перевірити, чи система реагує на зловмисні спроби доступу.

4. **Перевірка журналів Suricata:** Журнали Suricata зберігаються в папці /var/log/suricata/. Перевірте ці журнали, щоб оцінити, які загрози були виявлені і заблоковані:

cat /var/log/suricata/eve.json

Виконання лабораторної передбачає **три рівні складності** – базовий, середній та високий, що дозволить підвищувати свої навички щодо налаштування та конфігурації систем виявлення та запобігання вторгнень, а також здобути досвід в аналізі трафіку, виявленні загроз і застосуванні механізмів блокування атак у різних умовах безпеки мережі.

Рівень	Що потрібно зробити	Зміст звіту
базовий (60-74 балів)	Встановлення та базове налаштування Snort Встановлення та базове налаштування Suricata	- Опис процесу встановлення Snort та Suricata. - Опис процесу налаштування ПЗ. - Пояснення основних параметрів налаштування та конфігурації цих систем.
середній (75-89 балів)	**Виконати всі дії попереднього рівня.** Використання правил Snort та Suricata для конкретних сценаріїв. Тестування базового функціонування ПЗ. Перевірити реальну здатність систем виявляти загрози та реагувати на них. Аналіз результатів виявлення атак на основі стандартних правил.	- Перелік використаних правил. - Пояснення процесу додавання та налаштування правил для конкретних сценаріїв. - Опис результатів аналізу тестового трафіку. - Оцінка точності виявлення загроз
високий (90-100 балів)	**Виконати всі дії попереднього рівня.** Використання John the Ripper для перевірки паролів та виявлення вразливостей.	- Тестові сценарії, що використовувались для перевірки систем. - Оцінка результатів тестування. - Опис роботи з John the Ripper. - Результати тестування на виявлення вразливих паролів. - Рекомендації по оптимізації та підвищенню безпеки мережі.

Контрольні запитання:

1. Яка різниця між IDS та IPS?
2. Які методи використовують IDS/IPS для виявлення атак?

3. Як можна підвищити ефективність правил виявлення атак у Snort та Suricata?
4. Як аналізувати журнали IDS/IPS для виявлення потенційних загроз?
5. Які недоліки можуть мати системи IDS/IPS?
6. Що таке IDS та IPS? У чому різниця між ними?
7. Які основні завдання виконують системи IDS/IPS?
8. Які підходи до виявлення атак існують (сигнатурний, аномалійний, гібридний)?
9. Які обмеження та недоліки мають IDS/IPS-системи?
10. Що таке Snort і яке його призначення?
11. Які режими роботи підтримує Snort (sniffer, packet logger, NIDS/NIPS)?
12. Як виглядає базовий формат правила Snort (структура rule header та rule options)?
13. Які типи дій можуть виконуватись у правилах Snort (alert, log, pass, drop тощо)?
14. Де зберігаються журнали Snort і як їх можна переглянути?
15. Які джерела сигнатур можна використовувати для Snort?
16. Що таке Suricata і які її основні відмінності від Snort?
17. Які додаткові можливості Suricata (наприклад, багатопоточність, підтримка JSON-виводу)?
18. Які типи журналів генерує Suricata (eve.json, fast.log, stats.log)?
19. Як Suricata інтегрується з SIEM-системами (наприклад, Wazuh, ELK)?
20. Як виглядає приклад правила для Suricata?
21. Як перевірити, що Snort або Suricata коректно перехоплюють трафік?
22. Як відрізнити справжню атаку від хибнопозитивного спрацювання?
23. Які типи атак найчастіше виявляють Snort/Suricata (сканування портів, SQLi, DoS, brute-force)?
24. Які методи зниження кількості false positives у IDS/IPS?
25. Чому важливо оновлювати бази сигнатур Snort/Suricata?