ЛАБОРАТОРНА РОБОТА №4

СОРТУВАННЯ ТА РОЗСЛІДУВАННЯ ІНЦИДЕНТІВ

Мета роботи:

1. Ознайомитися з принципами сортування (тріажу) алертів та методами їх класифікації за пріоритетами.

2. Провести дослідження механізмів моніторингу та детекції загроз у Wazuh для аналізу безпеки хостів та мереж.

3. Розвинути навички складання звітів про інциденти та практичного застосування методології PICERL при розслідуванні подій безпеки.

Теоретичні відомості

Сортування (тріаж) алертів – це процес систематичної оцінки та класифікації попереджень безпеки для визначення їх пріоритету та необхідності подальшого розслідування. Це перший критичний крок у реагуванні на інциденти безпеки.

Рівні критичності алертів

Wazuh класифікує алерти за рівнями (0-15):

1. 0-4: Інформаційні події, зазвичай не потребують уваги.
2. 5-7: Низький пріоритет, потенційні проблеми.
3. 8-11: Середній пріоритет, потребують перевірки.
4. 12-15: Високий пріоритет, критичні події.

Методологія розслідування алертів

Методологія PICERL є стандартизованим підходом до управління інцидентами інформаційної безпеки. Вона забезпечує системність дій аналітика SOC, дозволяючи мінімізувати збитки та запобігти повторним атакам.

Методологія PICERL складається з наступних етапів:

1. Preparation (Підготовка) – розуміння інфраструктури.
2. Identification (Ідентифікація) – виявлення алерту.
3. Triage (Сортування) – оцінка і класифікація алерту.
4. Notification (Повідомлення) – інформування зацікавлених сторін.

5. Containment (Стримування) – обмеження поширення.
6. Eradication (Ліквідація) – усунення загрози.
7. Recovery (Відновлення) – повернення до нормальної роботи.
8. Lessons Learned (Висновки) – Документування та покращення.

Функціональні напрями моніторингу в системі Wazuh

Платформа Wazuh використовує декілька механізмів детекції для забезпечення комплексного захисту хостів та мережевої інфраструктури:

1. File Integrity Monitoring (FIM) – контроль цілісності критично важливих файлів та конфігурацій шляхом відстеження операцій створення, модифікації або видалення.

2. Log Analysis – агрегація та кореляція даних з журналів подій ОС та додатків, виявляючи приховані аномалії.

3. Vulnerability Detection – автоматичне виявлення вразливостей (пошук ПЗ із відомими CVE).

4. Intrusion Detection – активний пошук ознак експлуатації вразливостей, брутфорс-атак або присутності руткітів у реальному часі.

5. Policy Monitoring – порушення політик безпеки.

Класифікація результатів детекції

Для оцінки ефективності налаштованих правил моніторингу використовується матриця неточностей (Confusion Matrix), яка розділяє події на чотири категорії:

Легітимні спрацювання:

1. True Positive (TP) – правильне виявлення реальної загрози.

2. True Negative (TN) – коректне ігнорування системою безпечної активності.

Помилкові спрацювання:

3. False Positive (FP) – помилкова класифікація безпечної дії як загрози (вимагає більш точного налаштування правил).

4. False Negative (FN) – пропуск реальної загрози системою моніторингу (найбільш критичний показник, що вказує на недостатність заходів захисту).

Завдання на лабораторну роботу

Завдання 1. Запуск лабораторного середовища

1. Запустіть необхідні сервіси.

```
./lab-management.sh start wazuh
./lab-management.sh start vuln-lab
```

2. Підключіться до Wazuh Dashboard.

```
Відкрийте браузер: https://localhost:443
Логін: admin
Пароль: SecretPassword
```

Завдання 2. Генерація алертів File Integrity Monitoring.

1. У Kali Linux відкрийте термінал. Дізнайтесь IP-адресу контейнера Metasploitable.

```
sudo docker inspect metasploitable2 | grep IPAddress
```

2. Підключіться до контейнера metasploitable (значення параметра <IP-адреса> необхідно замінити на адресу, визначену в п. 1.).

```
ssh -oHostKeyAlgorithms=+ssh-rsa \
  -oPubkeyAcceptedAlgorithms=+ssh-rsa \
  msfadmin@<IP-адреса>
```

Пароль: msfadmin

3. Створіть підозрілий файл у директорії, що моніториться.

```
# Створіть файл
echo "suspicious content" > test_alert.conf


# Змініть вміст існуючого файлу
sed -i '/<\/ul>/a\
<h1>hacked</h1>\
<h2>try to return your access</h2>' vulnerable/twiki20030201/twiki-source/index.html

cat vulnerable/twiki20030201/twiki-source/index.html
```

```
 <title>Welcome to TWiki - A Web-based Collaboration Platform</title>
 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
</head>
<body bgcolor="#ffffff">
 <h1>Welcome to TWiki</h1>
 <ul>
  <li> <a href="readme.txt">readme.txt</a> </li>
  <li> <a href="license.txt">license.txt</a> </li>
  <li> <a href="TWikiDocumentation.html">TWikiDocumentation.html</a> </li>
  <li> <a href="TWikiHistory.html">TWikiHistory.html</a> </li>
  <li> Lets <a href="./bin/view/Main/WebHome">get started</a> with this web b
ased collaboration platform </li>
 </ul>
<h1> hacked </h1>
<h2>try to return your access</h2>
</body>
</html>
```

Рисунок 4 – Результат редагування файлу vulnerable/twiki20030201/twiki-source/index.html

```
# Видаліть тестовий файл
rm test_alert.conf
```

4. Перейдіть до Wazuh Dashboard та перейдіть до **Threat intelligence → Threat Hunting** у лівому меню.

5. Встановіть фільтр часу на Last 15 minutes.

6. Знайдіть алерти з Rule ID 550 (File added) або 553 (File modified).

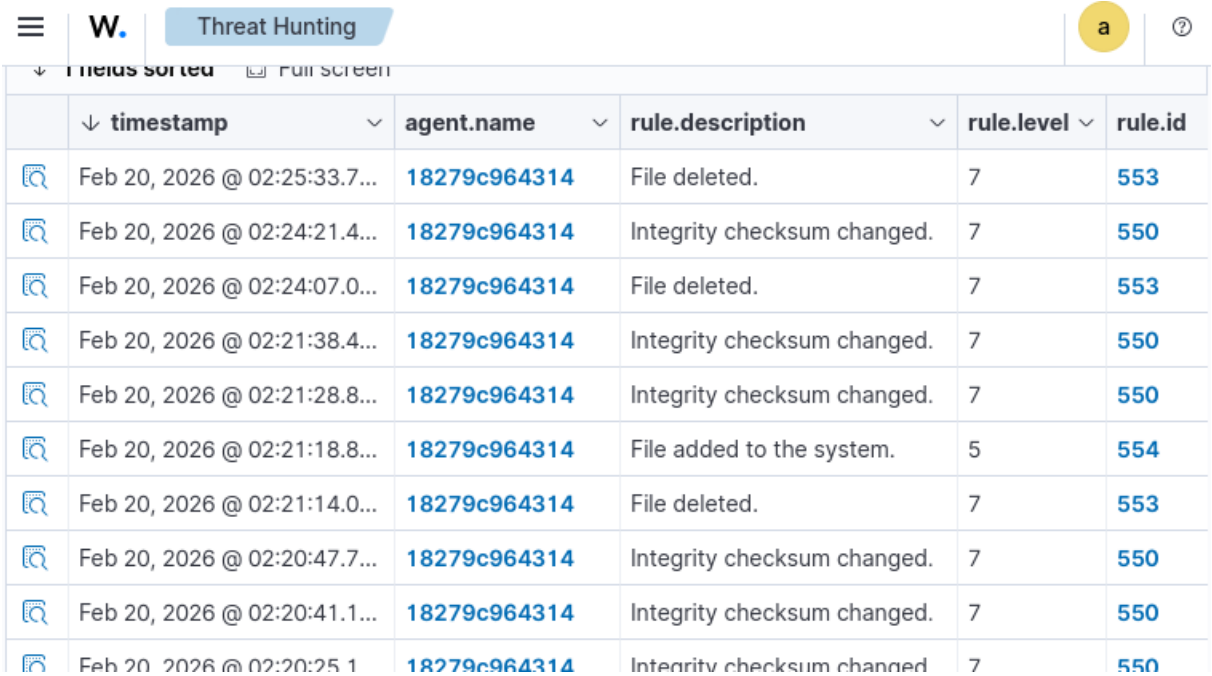

Рисунок 5 – Сповіщення Rule ID 550 або 553

7. Для одного алерту з кожної категорії (File deleted, Integrity checksum changed та File added to the system) необхідно задокументувати такі параметри події:

- Agent name: агент, який згенерував алерт.
- Rule description: опис правила.
- File path: шлях до зміненого файла.
- Timestamp: час події.
- MD5/SHA1: хеші файлу (якщо доступні).

Завдання 3. Генерація та розслідування алертів брутфорс-атак.

1. З Kali Linux згенеруйте невдалі спроби входу в DVWA.

```
# Створіть список паролів
echo -e "admin\npassword\n123456\ntest\ndvwa" > passwords.txt


#Запустіть brute force атаку на DVWA (2-3 рази)
hydra -l admin -P passwords.txt http-get://localhost/dvwa/login.php
```

2. Перейдіть до Wazuh Dashboard та оновіть вкладку **Threat Hunting**.

3. Знайдіть та проаналізуйте детальну інформацію про алерти з Rule ID 31101 або правила, пов'язані з "authentication failed".

	timestamp	agent.name	rule.description	rule.level	rule.id
	Feb 20, 2026 @ 02:30:48.4...	nginx-wazuh-agen	Web server 400 error code.	5	31101

Рисунок 6 – Правило з ID 31101

Завдання 4. Аналіз Web-атак через Nginx.

1. Увійдіть в обліковий запис на веб-сайті DVWA (*admin*/*password*) для отримання доступу до панелі з вразливими додатками (за потреби, натисніть на "*create/reset database*" та повторно увійдіть в обліковий запис):

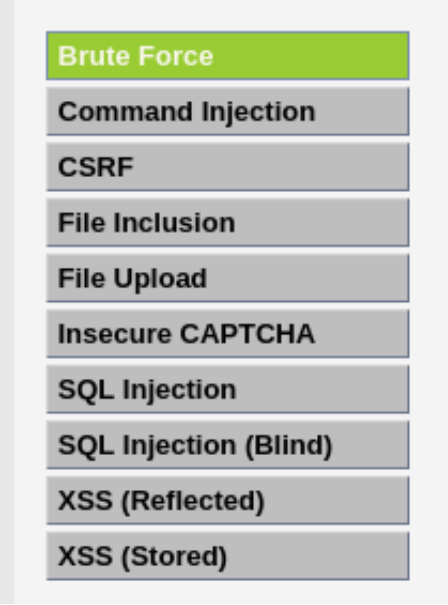

Рисунок 7 – Панель з вразливими додатками на веб-сайті DVWA

2. У розділі DVWA Security змініть поточний рівень безпеки з Impossible на Low.

3. Згенеруйте підозрілі web-запити з Kali (скопіюйте посилання у нову вкладку веб-браузера).

 # Спроба SQL Injection
http://localhost/vulnerabilities/sqli/index.php?id=1%27%20OR%20%271%27=%271&Submit=Submit

 # Спроба XSS
http://localhost/vulnerabilities/xss_r/index.php?name=%3Cscript%3Ealert(%27XSS%27)%3C/script%3E

 # Спроба Directory traversal
http://localhost/vulnerabilities/fi/?page= ../../../../../etc/passwd

4. Аналіз у Wazuh Dashboard. Перейдіть до **Threat Hunting → Events**.

5. Знайдіть сповіщення з Rule ID: 311**

6. Для кожного типу атаки визначте:

- Тип атаки (SQL Injection, XSS, Path Traversal).
- Source IP та User-Agent.
- HTTP-метод та URI.
- Response code.
- Чи була атака успішною?

Завдання 5. Створення звіту про розслідування.

Для **одного з виявлених алертів** заповніть звіт відповідно до наведеної структури:

1. Базова інформація:

- Alert ID: [з Wazuh]

- Timestamp: [дата та час]

- Rule ID and Description:

- Severity Level:

2. Деталі події:

- Source: [IP/Host/Agent]

- Destination: [IP/Port/Service]

- Action taken: [що було зроблено]

- Affected systems:

3. Аналіз:

- Classification: [True Positive]

- Attack vector:

- Indicators of Compromise (IoCs):

- Related events:

4. Impact assessment:

- Criticality: [High/Medium/Low]

- Data compromised: [Yes/No/Unknown]

- Systems affected:

- Business impact:

5. Response actions:

- Immediate actions taken:

- Containment measures:

- Recommendations:

6. Timeline

- Detection time:

- Response time:

- Resolution time:

Завдання 6. Експортуйте алерт з Wazuh.

Для обраного алерту виконайте одну з наступних дій (необхідно для виконання завдання 7):

Виберіть алерт у Dashboard

Натисніть кнопку експорту (JSON)

Збережіть для документації

АБО

Перегляньте детальну інформацію про алерт у форматі JSON напряму на сторінці Threat Hunting

Завдання 7. Створення базової SOP (Standard Operating Procedure).

Створіть простий SOP-документ для тріажу алертів. Пройдіть через цю процедуру з одним з ваших алертів і заповніть чеклист.

SOP: ALERT TRIAGE PROCEDURE

1. INITIAL ASSESSMENT.

☐ Оцінити рівень серйозності алерту

☐ Визначити агента/систему, на якій зафіксовано подію.

☐ Перевірити частоту спрацювань

2. INFORMATION GATHERING.

☐ Зібрати деталі події з Wazuh (Rule ID, опис, лог)

☐ Перевірити пов'язані події (±15 хвилин)

☐ Проаналізувати source/destination IP

3. ANALYSIS.

☐ Визначити тип атаки або аномалії

☐ Встановити, чи є подія True Positive або False Positive

☐ Коротко обгрунтувати рішення (1-2 речення).

4. DECISION.

☐ *Визначити рівень критичності (Low / Medium/ High).*

☐ *Запропонувати базові заходи реагування.*

☐ *Зафіксувати прийняте рішення у звіті.*

Завдання 8. Зупинка середовища.

```
./lab-management.sh stop wazuh
./lab-management.sh stop vuln-lab
```

Контрольні запитання

1. Що означає тріаж алертів у SOC?

2. Який рівень серйозності в Wazuh вказує на критичні події?

3. До якого рівня належать події, що вимагають перевірки?

4. Що таке False Positive?

5. Який етап методології PICERL відповідає за обмеження поширення загрози?

6. Який етап методології PICERL відповідає за виявлення алерту?

7. Що відслідковує FIM у системі Wazuh?

8. Який тип спрацювання вказує на пропуск реальної загрози?

9. Що вказується у полі Response time?

10. Що таке True Negative (TN) у системі моніторингу?

11. Який етап PICERL відповідає за документування та покращення процесу?

12. Який рівень серйозності вказує на інформаційнні події, зазвичай не потребуючи уваги?