МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Державний університет «Житомирська політехніка»
Кафедра комп'ютерної інженерії та кібербезпеки

ЗВІТ
З ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

Виконав:
студент 2-го курсу групи КБм-22-1
спеціальності 125 «Кібербезпека»
_____ Гончаров М. В.
Керівник:
_____ Воротніков В. В.,
д.т.н., доцент, професор кафедри КІ та КБ,

Оцінка:
Національна шкала _____
Кількість балів _____
ECTS _____

Голова комісії:

_____ Воротніков В. В.,
д.т.н., доцент, професор кафедри КІ та КБ

Члени комісії:

_____ Єфіменко А. А.,
к.т.н. доцент, завідувач кафедри КІ та КБ
_____ Росінський Ю. М.,
к.т.н. доцент, доцент кафедри КІ та КБ
_____ Шелуха О. О.,
к.т.н., доцент кафедри КІ та КБ
_____ Колощук М. С.,
асистент кафедри КІ та КБ

м. Житомир – 2023 рік

ЗМІСТ

ВСТУП

Зважаючи на теперішнє складне економічне положення українських компаній та і світових компаній загалом, спричинене війною та пандемією COVID-19, багатьом з них довелось відмовитись від утримування власного операційного центру безпеки (далі SOC). Більшість рішень, наявних на ринку є дороговартісними, а їх різні плани ліцензіювання можуть бути не підлаштованими до потреб певного конкретного SOC. Це призводить скорочення працівників SOC та до перенесення їх обов'язків на працівників IT-відділів та їх інструменти моніторингу, які через свою специфику можуть не вловлювати всіх подій та інцидентів інформаційної безпеки. Однак, сучасний перелік продуктів на ринку безпекових рішень також містить широкий вибір безкоштовних продуктів, які можуть бути скомпоновані в надійний та зручний SOC на малих та середніх підприємствах.

Метою даної переддипломної практики є вивчення та аналіз безкшотовних відкритих рішень для побудови SOC, що в свою чергу включає проєктування та побудову в віртуальному середовищі такого SOC. Практика спрямована на розвиток компетентностей, визначених стандартом вищої освіти зі спеціальності 125 "Кібербезпека", а саме: здатності застосовувати знання у практичних ситуаціях, проводити дослідження на відповідному рівні та оцінювати якість виконуваних робіт, а також здатності аналізувати, розробляти і супроводжувати систему управління інформаційною безпекою та/або кібербезпекою організації.

У розділі 1 "Загальна характеристика теми переддипломної практики" розглянуто та проаналізовано сучасні безкоштовні відкриті безпекові рішення, механізм роботи SOC на основі обраної комбінації рішень, методологію використану у ході дослідження та використані інструменти. В розділі 2 "Проєктування підсистеми захисту та моніторингу подій безпеки" створено топологічну схему мережі, надано сценарії налаштування мережевого обладнання, файли конфігурацій безпекових рішень та кроки по їх інтеграції у функціонуючий SOC.

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ СУЧАСНИХ БЕЗПЕКОВИХ РІШЕНЬ ДЛЯ ПОБУДОВИ SOC

1.1 Огляд та аналіз сучасних безкоштовних відкритих безпекових рішень

В сучасному цифровому світі, де кожну хвилину обробляється та передається велика кількість інформації, питання забезпечення безпеки даних набуває особливого значення. Велика кількість загроз інформаційній безпеці вимагає вдосконалення заходів захисту, а технологічний прогрес пропонує нові можливості в цьому напрямі. Однією з альтернатив традиційним комерційним засобам забезпечення інформаційної безпеки є використання відкритих безкоштовних рішень (далі FOSS, Free Open Source Software). У даному підрозділі буде проведено огляд та аналіз сучасних безкоштовних відкритих засобів для забезпечення інформаційної безпеки.

FOSS базуються на принципах відкритості програмного коду та можливості вільного розповсюдження та модифікації. Це дозволяє спільноті розробників активно співпрацювати в розробці та удосконаленні засобів забезпечення безпеки.

Однією з головних переваг використання FOSS є доступність вихідного коду. Це дозволяє індивідуальним користувачам, організаціям та спільноті розробників перевіряти, адаптувати та вдосконалювати програмне забезпечення відповідно до власних потреб.

За останні роки ринок безпекових FOSS збагатився великою кількістю нових рішень, які пропонують гідну альтернативу вже наявним комерійним рішеннями від таких відомих світових компаній-гігантів, як Microsoft, Amazon, Google, IBM Cisco, Fortinet та Palo Alto. В ході проходження практики було прийнято рішення про проєктування лише базового SOC, з компонентів без яких SOC буде просто напросто не ефективний. Саме компоненти SIEM та SOAR було обраними базою проектованого SOC. Компоненти по типу XDR, IDPS, пісочниць, Honeypot системи та сканеру вразливостей були винесені для інтеграції в спроектований SOC в ході виконання магстрської роботи. Серед FOSS рішень SIEM та SOAR були проаналізовані наступні:

1. **SIEM**. Security Information and Event Management (SIEM) - це програмне забезпечення, яке підвищує рівень безпеки IT-середовища, поєднуючи управління інформацією про безпеку (SIM) та управління подіями безпеки (SEM). Рішення SIEM покращують виявлення загроз, дотримання нормативних вимог та управління інцидентами безпеки шляхом збору та аналізу даних про події безпеки в реальному часі та історичних даних і джерел:

– ***AlienVault OSSIM.*** AT&T Cybersecurity надає AlienVault OSSIM, інструмент SIEM з відкритим вихідним кодом, заснований на рішенні USM від AlienVault. AlienVault OSSIM об'єднує багато проектів з відкритим вихідним кодом в єдиний пакет, близький до вищезгаданих продуктів. AlienVault OSSIM також має функції відстежувати та аудиту додатків.

– ***ELK Stack***. До складу рішення ELK Stack входять безкоштовні SIEM-продукти. Наприклад, ELK може компілювати журнали майже з усіх джерел даних за допомогою вбудованих компонентів Logstash. Таким чином, ці дані журналів можуть бути об'єднані в широкому спектрі плагінів, хоча ручні правила безпеки є необхідними.

– ***SIEMonster*** пропонує як безкоштовне SIEM, так і платне рішення. Як і у випадку з багатьма іншими рішеннями, фреймворк SIEMonster пропонує централізований інтерфейс управління інструментами для аналізу даних, розвідки загроз і різного програмного забезпечення з відкритим вихідним кодом. На відміну від деяких інших рішень SIEM з відкритим вихідним кодом, цей продукт розгортається в хмарі.

2. **SOAR**. Система оркестрування, автоматизації та реагування на загрози безпеці (SOAR) - це набір сумісних програм, які дозволяють організації збирати дані про загрози безпеці та реагувати на події безпеки з мінімальною участю людини або взагалі без неї. Метою використання платформи SOAR є підвищення ефективності операцій з фізичної та цифрової безпеки:

– ***TheHive Project***. TheHive – це масштабована платформа реагування на інциденти безпеки, тісно інтегрована з MISP (платформою обміну інформацією про шкідливе програмне забезпечення), покликана полегшити життя SOC, CSIRT,

CERT і будь-яким фахівцям з інформаційної безпеки, які мають справу з інцидентами безпеки, що потребують швидкого розслідування та реагування.

– ***Shuffle*** – це SOAR з відкритим вихідним кодом. Він має на меті надати всі можливості, необхідні для передачі даних по всьому підприємству, за допомогою додатків, що працюють за принципом "підключи і працюй", роблячи автоматизацію доступною для кожного.

– ***Walkoff*** – гнучка, проста у використанні система автоматизації, що дозволяє користувачам інтегрувати свої можливості та пристрої, щоб позбутися повторюваних, нудних завдань, які сповільнюють роботу.

1.2 Механізм роботи SOC на основі обраних FOSS рішень

В результаті проведеного аналізу, в якості SIEM був обраний ELK Stack, а в якості SOAR комбінація TheHive + Cortex + Shuffle.

ELK Stack, за рахунок широкого вибору агентів збирання журналів, дозволить збирати інформацію та події з всіх необхідних пристроїв та додатків незалежно від їх вендора. На основі цих даних аналітики зможуть реалізувати набір правил, який буде виявляти підозрілі та зловмисні дії. Після цього порушення, створенні в результаті спрацювання описаних правил, будуть передані TheHive.

TheHive, як система реагування на інциденти, дозволить аналітикам провести розслідування порушення в зручному і зрозумілому режимі. В процесі розслідування аналітики зможуть доповнювати наявні індикатори компрометації (IP адреси, хеші файлів, URL посилання і т.д.) виділені з порушення за допомогою аналізаторів (analyzers) компоненту Cortex. Після розслідування, аналітики матимуть змогу відреагувати на тей чи інший інцидент за допомогою відповідачів (responders) компоненту Cortex, наприклад заблокувавши IP-адресу зловмисника на мережевому екрані. Деякі з рутинних задач, наприклад запуск сканування мережі з подальшим створенням кейсу з результатами в TheHive, зможуть бути автоматизовані через Shuffle.

Якщо говорити про вектори розбудови такої базової платформи в повноцінний SOC зі всіма необхідними компонентами, однією з можливих реалізацій є SOC, що включає наступні компоненти:

1. SIEM система – ELK Stack
2. SOAR – TheHive + Cortex + Shuffle
3. Система IPS/IDS – Snort
4. EDR/XDR – Wazuh
5. Розвідка загроз – MISP
6. Сканер вразливостей – OpenVAS
7. Аналіз шкідливого ПЗ – YARA
8. Рішення пасток (Deception/Honey pot) – Honeyd.
9. Тестування на виявлення – Atomic Red Team.

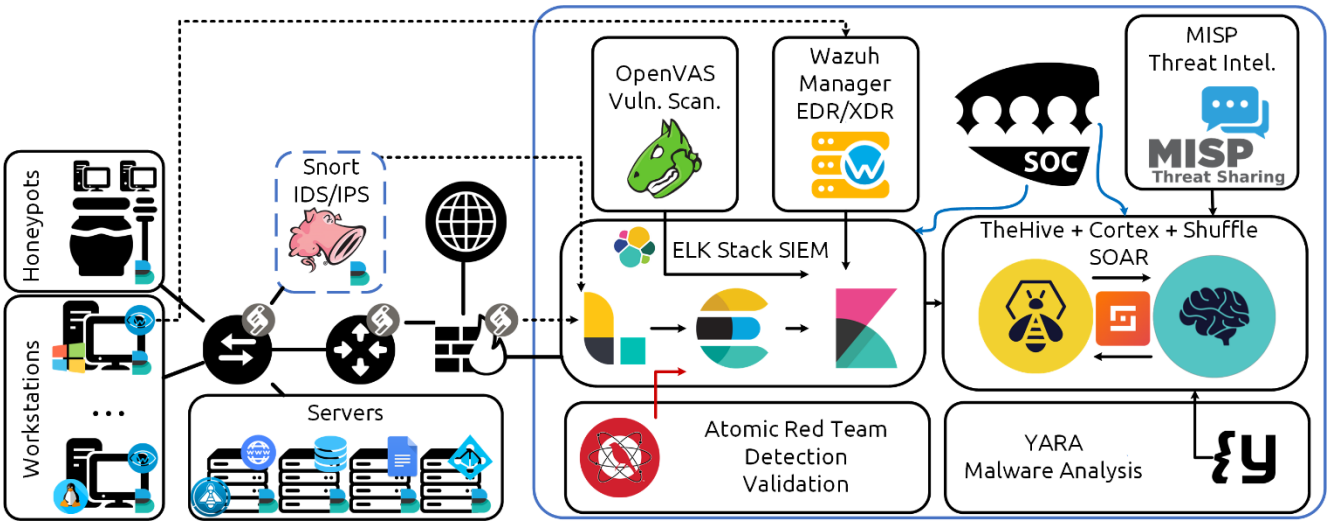

Рисунок 1.2.1 – Схематична діаграма SOC на вище

У такому SOC компоненти взаємодіють наступним чином:

– **SIEM** (ELK Stack) отримує дані журналів від агентів Wazuh і filebeat, встановлених на кінцевих точках. Він також отримує сповіщення від Snort, Wazuh та Honeydeal.

– **SOAR** (TheHive + Cortex) отримує сповіщення від SIEM і автоматично запускає плейлисти реагування на інциденти за допомогою Cortex.

– **IPS/IDS** (Snort) відстежує мережевий трафік і надсилає сповіщення до SIEM, якщо виявляє підозрілу або зловмисну активність.

– **EDR/XDR** (Wazuh) захищає кінцеві точки, постійно відстежує і повідомляє про будь-яку підозрілу активність в SIEM.

– **Розвідка загроз** (MISP) надає контекст подій безпеки та доповнює оповіщення, що генеруються SIEM.

– **Сканер вразливостей** (OpenVAS) сканує мережу та кінцеві точки на наявність вразливостей і повідомляє про результати в SIEM.

– **Рішення пасток** (Honeyd) використовується для виявлення та затримання зловмисників шляхом емуляції вразливих систем та сервісів.

– **Аналіз шкідливого ПЗ** (YARA) використовується для аналізу підозрілих файлів, виявлених агентами Wazuh та скануванням Open-VAS.

– **Тестування на виявлення** (Atomic Red Team) забезпечує безперервне тестування можливостей виявлення SOC, щоб гарантувати, що вона ефективно протидіє новим загрозам.

SOAR на базі TheHive + Cortex чудово інтегрується з усіма цими рішеннями і може бути використаний для автоматизації процесів наступних SOC-рішень:

– **YARA**: Автоматизуйте сканування YARA вхідних файлів або семплів. Наприклад, коли новий файл надсилається до TheHive, Cortex може автоматично запускати проти нього правила YARA, щоб визначити, чи відповідає він будь-яким відомим індикаторам шкідливого програмного забезпечення або загроз.

– **OpenVAS**: Автоматизація сканування вразливостей і створення звітів за допомогою OpenVAS. Наприклад, Cortex можна використовувати для сканування певних об'єктів або груп об'єктів, а TheHive генеруватиме сповіщення та тікети для всіх знайдених вразливостей.

– **Atomic Red Team**: Автоматизація тестування виявлених вразливостей за допомогою фреймворку тестування Atomic Red Team. Наприклад, Cortex можна використовувати для запуску певних тестів на кінцевій точці або групі кінцевих

точок, а TheHive - для генерації сповіщень і тікетів (кейсів) для будь-яких знайдених вразливостей.

– **MISP**: Автоматизація обміну та кореляції даних про індикатори компрометації. Наприклад, Cortex можна використовувати для запиту до MISP щодо індикаторів загроз, пов'язаних з поточним розслідуванням, а TheHive потім співвідносе ці індикатори з іншими подіями або оповіщеннями в системі.

1.3 Методологія дослідження та використані інструменти

У цьому підрозділі описана методологія, яку було використано під час проведення проєктування SOC на базі FOSS, а також наведені основні інструменти, що допомогли в реалізації цієї методології. Вищевказані аспекти мають важливе значення для забезпечення наукової обґрунтованості та достовірності отриманих результатів.

Методологія дослідження

При розробці методології дослідження було враховано комплексність та специфіку теми проєктування SOC на базі безкоштовних відкритих рішень. Дослідження були поділено на наступні етапи:

Аналіз літературних джерел: Перший етап передбачав збір та аналіз наукової літератури та публікацій, що стосуються наявних безкоштовних безпекових рішень.

Проєктування мережевої інфраструктури: На цьому етапі було спроєктовано мережеву інфраструктуру SOC, що має відкритий доступ в Інтернет, складається з мережевого обладнання у вигляді маршрутиразторів та комутаторів, які зв’язують пристрої зі встановленими безпековими рішеннями у діючий SOC.

Налаштування та перевірка зв’язності та функціонування SOC: Цей етап відповідав за налаштування мережевого обладнання та безпекових рішень та їх з’єднання між собою шляхом налаштування конфігураційних файлів та графічного інтерфейсу.

Використані інструменти

У ході проходження практики були використані наступні інструменти:

Документаційні засоби: Використання наукових статей, документації та офіційних джерел дозволило отримати актуальну та достовірну інформацію.

Емуляційні платформи: Для проєктування мережної інфраструктури SOC. були використані наступні платформи: GNS3, Oracle VM Virtualbox та Microsoft Hyper-V.

Безпекові рішення:

- у якості SIEM – Elastic Stack (Elasticsearch, Kibana, Logstash);
- у якості SOAR – поєднання TheHive + Cortex + Shuffle.

РОЗДІЛ 2. ПРОЄКТУВАННЯ SOC НА БАЗІ FOSS РІШЕНЬ

2.1 Проєктування мережної інфраструктури SOC

Корпоративна мережа – це логічно відокремлена група комп'ютерів, маршрутизаторів та інших частин IT-інфраструктури, які функціонують поза традиційними межами Інтернету. Її часто називають Інтранет. Термін Інтранет описує мережу, яка, на відміну від Інтернету, призначена для доступу лише для певної групи

людей.

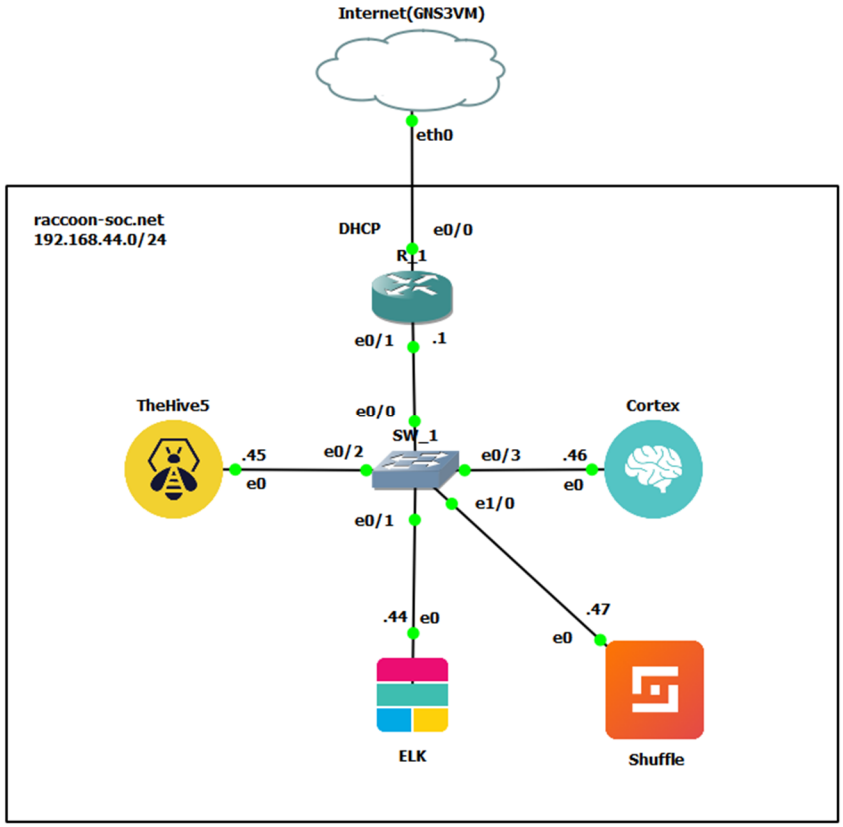

Рисунок 2.1.1 – Топологічна сехма мережі SOC

Слід зазначити, що цей сценарій було проемульовано за допомогою ПЗ Oracle VirtualBox, GNS3 та Hyper-V. Отож для емуляції мережної інфраструктури використовувались домашній маршрутизатор (Mercusys AC12G) та ноутбук (Windows 11 x64 Enterprise, 32 ГБ ОЗП). Для зручності налаштування пристроїв та серверів мережі, ВМ GNS3 під'єднано до мережі використовуючи підключення міст, яке закріплено за мережевим інтерфейсом Ethernet ноутбука. Це дозволяє ВМ отримати IP-адресу, як пристрою домашньої мережі (192.168.1.0/24).

Таблиця 2.1.1 – Параметри інтерфейсів пристроїв

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до інтерфейсу
Internet(GNS3VM) (вузол Cloud)	Eth0	Маршрутизатор R1	E0
Маршрутизатор R1 (adventerprisek9-15.4.2T4)	E0/0	Internet(GNS3VM)	Eth2
	E0/1	Комутатор SW1	E0/0

Комутатор SW1 (adventerprise-15.2c)	E0/0	Маршрутизатор R1	E0/1
	E0/1	Сервер ELK	E0
	E0/2	Сервер TheHive	E0
	E0/3	Сервер Cortex	E0
	E1/0	Сервер Shuffle	E0
Сервер ELK (Ubuntu Server 22.04)	E0	Комутатор SW1	E0/1
Сервер TheHive (Ubuntu Server 22.04)	E0		E0/2
Сервер Cortex (Ubuntu Server 22.04)	E0		E0/3
Сервер Shuffle (Ubuntu Server 22.04)	E0		E1/0

Таблиця 2.1.2 – Параметри IP-адресації мережі

Мережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска	/xx
Підмережа raccoon-soc.net	–	192.168.44.0	255.255.255.0	/24
Маршрутизатор R1	Інтерфейс E0/0	DHCP		
	Інтерфейс E0/1	192.168.44.1	255.255.255.0	/24
Комутатор SW1	Інтерфейс Vlan 1	192.168.44.100	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.44.1	–	–
Сервер ELK	Мережний адаптер	192.168.44.44	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.44.1	–	–
Сервер TheHive	Мережний адаптер	192.168.44.45	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.44.1	–	–
Сервер Cortex	Мережний адаптер	192.168.44.46	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.44.1	–	–
Сервер Shuffle	Мережний адаптер	192.168.44.47	255.255.255.0	/24
	Шлюз за замовчуванням	192.168.44.1	–	–

Сценарії налаштування параметрів IP-адресації, протоколу віддаленого керування SSH та протоколу мережевого часу NTP на маршрутизаторі **R1**:

R1:

```
Router>en
Router#conf t
Router(config)#hostname R1
R1(config)#username admin privilege 15 algorithm-type sha256 secret123!@#
R1(config)#int e0/0
R1(config-if)#ip address dhcp
R1(config-if)#no shut
R1(config-if)#int e0/1
R1(config-if)#ip address 192.168.44.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.1
R1(config)#ip domain-name raccoon-soc.net
R1(config)#ip dns server
R1(config)#ip name-server 192.168.1.1
R1(config)#crypto key generate rsa general-key modulus 2048
R1(config)#ip ssh version 2
R1(config)#line vty 0 4
R1(config-line)#logging synchronous
R1(config-line)#login local
R1(config-line)#exec-timeout 20
R1(config-line)#transport input ssh
R1(config-line)#exit
R1(config)#ntp server 0.ua.pool.ntp.org
R1(config)#clock timezone EET +2
R1(config)#clock summer-time EEST reccuring
R1(config)#do write
```

```
> ssh admin@192.168.44.1
(admin@192.168.44.1) Password:

R_1#show ip int br
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                192.168.1.108   YES DHCP   up                    up
Ethernet0/1                192.168.44.1    YES NVRAM  up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down down
Ethernet0/3                unassigned      YES NVRAM  administratively down down
Serial1/0                  unassigned      YES NVRAM  administratively down down
Serial1/1                  unassigned      YES NVRAM  administratively down down
Serial1/2                  unassigned      YES NVRAM  administratively down down
Serial1/3                  unassigned      YES NVRAM  administratively down down
```

Рисунок 2.1.2 – Результат виконання команди ***show ip interface brief*** на маршрутизаторі ***R1***

```
R_1#show ip route | section e Codes

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.1.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.108/32 is directly connected, Ethernet0/0
      192.168.44.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.44.0/24 is directly connected, Ethernet0/1
L        192.168.44.1/32 is directly connected, Ethernet0/1
```

Рисунок 2.1.3 – Результат виконання команди ***show ip route*** на маршрутизаторі ***R1***

```
R_1#show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256
-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC3pzXB778XgHTRCfsn5fuwBGnRLsVukW9mdZw5MlEq
xfWCVYP+eBzfLpdJsgzW8vHpxNlS+VAAuOMlUhk0KlddcLJKJnYDof4Ii7FQF2jsB9ohC1ijjJ3AQBTx
DKGQeZAKSziF3Iv/CzjyeApJKe/yZ0hbZgiWJu0kDUm52q3iERifZuuho3M7EhTLTV3BGEny6NUb0h5Y
RsV+8ninSa+q43pDh7J1jZkiUm3gOSLfWACmH227spMLtkn+Lmcq/pqnjA+0m4ZG09ks2Rx76tzgj1vf
cpMAPSeMZreSq/+7o4RlMs1uxeSHol84Zhsg+a+XbyYWCEBQvIhIhmYswMIn
```

Рисунок 2.1.4 – Результат виконання команди ***show ip ssh*** на маршрутизаторі R2

```
R_1#show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~82.193.104.168  212.160.106.226  2    226    256   377  4.906   2.469  4.688
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R_1#show clock detail
20:27:52.071 EET Sun Nov 26 2023
Time source is NTP
Summer time starts 02:00:00 EET Sun Mar 10 2024
Summer time ends 02:00:00 EEST Sun Nov 3 2024
```

Рисунок 2.1.5 – Результат виконання команд show ntp associations та show clock detail на маршрутизаторі R1

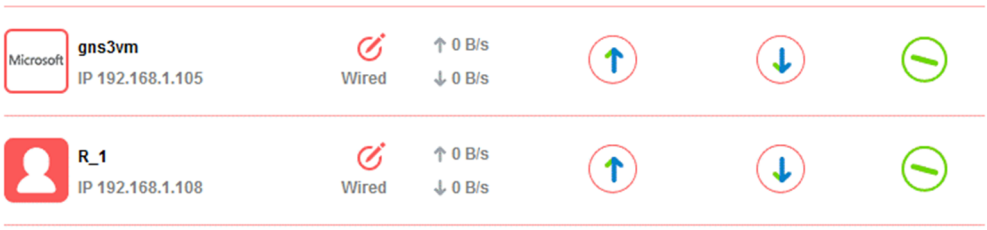

Рисунок 2.1.6 – Перелік пристроїв, використаних у віртуальній підмережі, що отримали DHCP-параметри та є підключеними до домашньої мережі

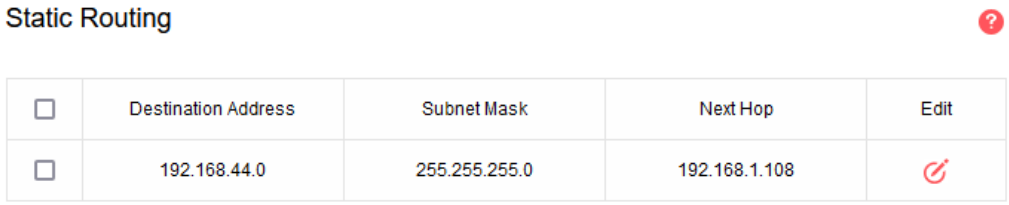

Рисунок 2.1.7 – Статична маршрутизація для віртуальної підмережі в меню налаштування маршрутизації домашнього маршрутизатора

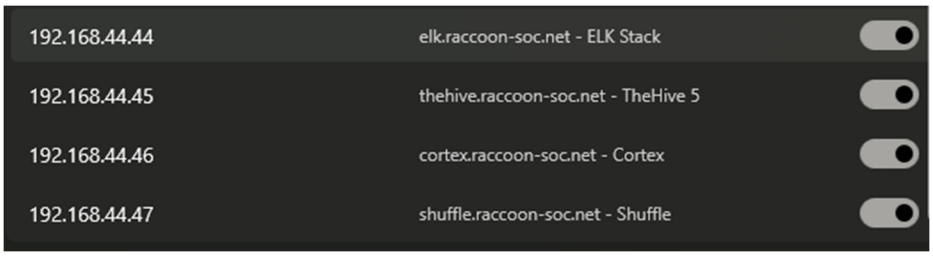

Рисунок 2.1.8 – Записи у файлі ***hosts***

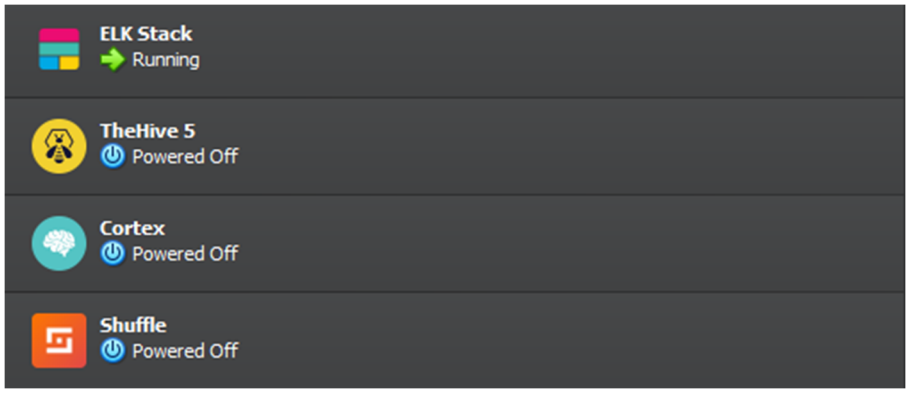

Рисунок 2.1.9 – ВМ в Orcale VM VirtualBox

Сценарії налаштування параметрів IP-адресації, протоколу віддаленого керування SSH та протоколу мережевого часу NTP на комутаторі ***SW1***:

SW1:

```
Switch>en
Switch#conf t
Switch(config)#hostname SW1
SW1(config)#username admin privilege 15 algorithm-type sha256 secret 123!@#
SW1(config)#int vlan 1
SW1(config-if)#ip address 192.168.44.100 255.255.255.0
SW1(config-if)#ntp broadcast client
SW1(config-if)#exit
SW1(config)#ip default-gateway 192.168.44.1
SW1(config)#ip domain-name raccoon-soc.net
SW1(config)#ip name-server 192.168.44.1
SW1(config)#ip ssh version 2
SW1(config)#crypto key generate rsa general-key modulus 2048
SW1(config)#line vty 0 4
SW1(config-line)#logging synchronous
SW1(config-line)#login local
SW1(config-line)#exec-timeout 20
SW1(config-line)#transport input ssh
SW1(config-line)#exit
SW1(config)#ntp server 192.168.44.1
SW1(config)#clock timezone EET +2
SW1(config)#clock summer-time EEST reccuring
```

```
SW1(config)#do write
```

```
SW_1#show ip int br | i Vlan1
Vlan1                  192.168.44.100  YES NVRAM  up                    up
SW_1#show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-
cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): SW_1.raccoon-soc.net
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCvwpClvDSy3Il6eZDwk9JSD1c+Ey/BnqiWD2vvxOHv
SZPMpzk6/eUPlMoYUtMdom5jLopH911LuP9KwluobSHS5ofseCfmCNKUj1u8aQxVenkzMxh3n4Lzp7Ap
jMqG0WNXf3X4lW1XBYKyiVlu2CMV33p0PM4cap5H4fm+okyEsLbQOPhptJFdi7ICqjspy+nVlnewElTh
nMFVTNQbOaOnHWzszCJoii6vI0YWqnj644bmNE8DznarciBdSGzBQMpvSLilI0lOU0FyQ2+44oZC6XK2
49Ia3klZRCYVOaxajKKXuNtFd8fLAMCDvLl/Pr86ZwSQhsIsAi+30zCP6UG/
SW_1#show ntp assoc

  address         ref clock       st   when   poll reach  delay  offset   disp
*~192.168.44.1    82.193.104.168   3    887   1024   377  0.000   0.000  2.074
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
SW_1#show clock detail
23:27:51.564 EET Sun Nov 26 2023
Time source is NTP
Summer time starts 02:00:00 EET Sun Mar 10 2024
Summer time ends 02:00:00 EEST Sun Nov 3 2024
```

Рисунок 2.15 – Результат виконання команд show ip interface brief, show ip ssh, show ntp associations та show clock detail на комутаторі SW1

2.2 Початкові налаштування SIEM та SOAR

Так як всі обрані рішення запущені на серверах з ОС Linux Ubuntu Server 22.04, їх налаштування будуть виконуватись шляхом зміни конфігураційних файлів у каталозі ***/etc***, що є типовим для ОС Linux.

```
root@elk-stack:~# cat /etc/elasticsearch/elasticsearch.yml | grep -v '^#'
cluster.name: thp
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 192.168.44.44
http.port: 9200
discovery.type: single-node
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true

xpack:
  security:
    authc:
      realms:
        native:
          native1:
            order: 0

script.allowed_types: inline,stored
```

Рисунок 2.2.1 – Налаштування компоненту ELK Elasticsearch у файлі

/etc/elasticsearch/elasticsearch.yml

```
root@elk-stack:~# cat /etc/kibana/kibana.yml | grep -v '^#' | grep -v -e '^$'
server.port: 5601
server.host: "192.168.44.44"
elasticsearch.hosts: ["http://192.168.44.44:9200"]
elasticsearch.username: "kibana_system"
elasticsearch.password: "123!@#"
xpack.security.enabled: true
xpack.ingestManager.fleet.tlsCheckDisabled: true
xpack.encryptedSavedObjects.encryptionKey: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

Рисунок 2.2.2 – Налаштування компоненту ELK Kibana у файлі

/etc/kibana/kibana.yml

```
kilroy@thehive:~$ sudo cat /etc/thehive/application.conf | grep -v -e '^\s*#' | grep -v -e '
^$'
http.address = 192.168.44.45
http.port = 9000
include "/etc/thehive/secret.conf"
db.janusgraph {
  storage {
    backend = cql
    hostname = ["127.0.0.1"]
    cql {
      cluster-name = thp
      keyspace = thehive
    }
  }
  index.search {
    backend = elasticsearch
    hostname = ["192.168.44.44"]
    index-name = thehive
    elasticsearch {
      http {
        auth {
          type = basic
          basic {
            username = elastic
            password = "123!@#"
          }
        }
      }
    }
  }
}
storage {
  provider = localfs
  localfs.location = /opt/thp_data/files/thehive
}
play.http.parser.maxDiskBuffer = 1GB
play.http.parser.maxMemoryBuffer = 10M
play.http.context = "/"
scalligraph.modules += org.thp.thehive.connector.cortex.CortexModule
scalligraph.modules += org.thp.thehive.connector.misp.MispModule
```

Рисунок 2.2.3 – Налаштування компоненту SOAR TheHive 5 у файлі

/etc/thehive/application.conf

Рисунок 2.2.4 – Налаштування компоненту SOAR TheHive 5 у файлі ***/etc/cortex/application.conf***

Подальші налаштування цих рішень проходять всередині графічного інтерфейсу. Файлу налаштувань Shuffle не подано, адже саме рішення запускається в якості Docker контейнеру та працює «з коробки». Також слід зазначити, що в якості сховища TheHive та Cortex використовують elasticsearch, а так як він є компонентом нашої SIEM, саме він вказаний у конфігураційних файлах.

2.3 Перевірка зв’язності між компонентами SOC

TheHive та Cortex тісно взаємодію між собою, але для налаштування цієї взаємодії потрібно створити відповідного користувача в Cortex та за допомогою його API-токену підключити Cortex до TheHive.

Рисунок 2.3.1 – Користувачі організації Cloud Native Solutions в TheHive

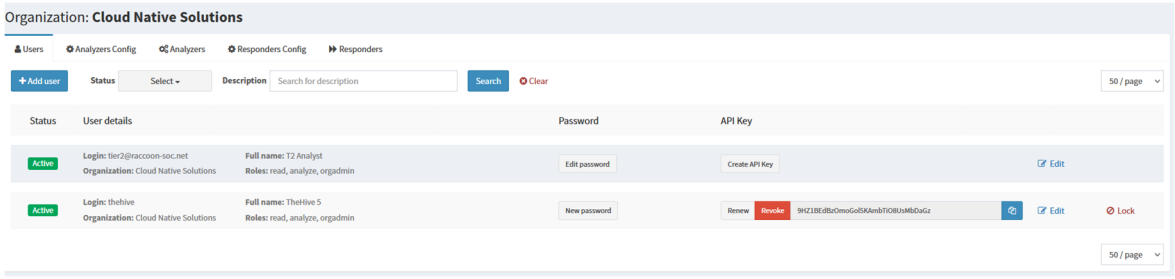

Рисунок 2.3.2 – Користувачі організації Cloud Native Solutions в Cortex

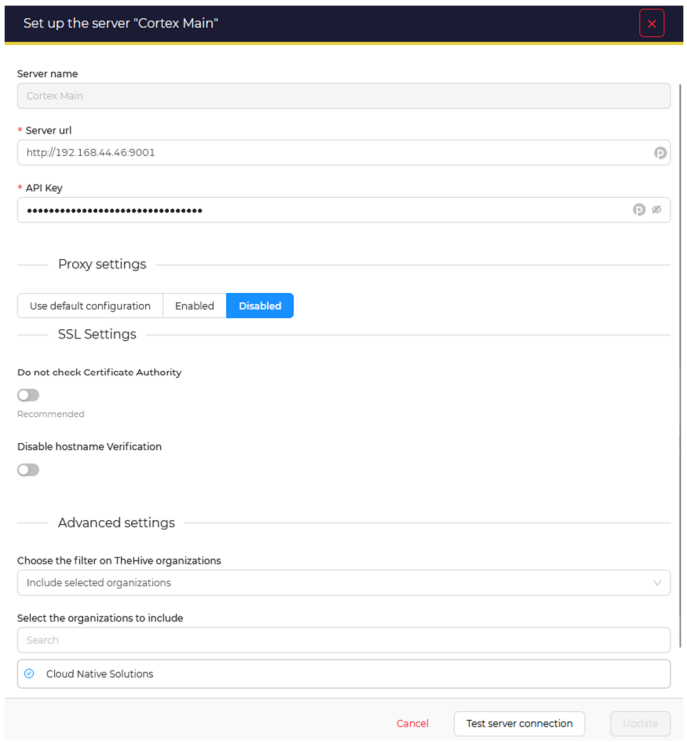

Рисунок 2.3.3 – Підключення Cortex до TheHive

Після успішного здійснення підключення можемо побачити активовані аналізатори (analyzers) Cortex в налаштуваннях TheHive.

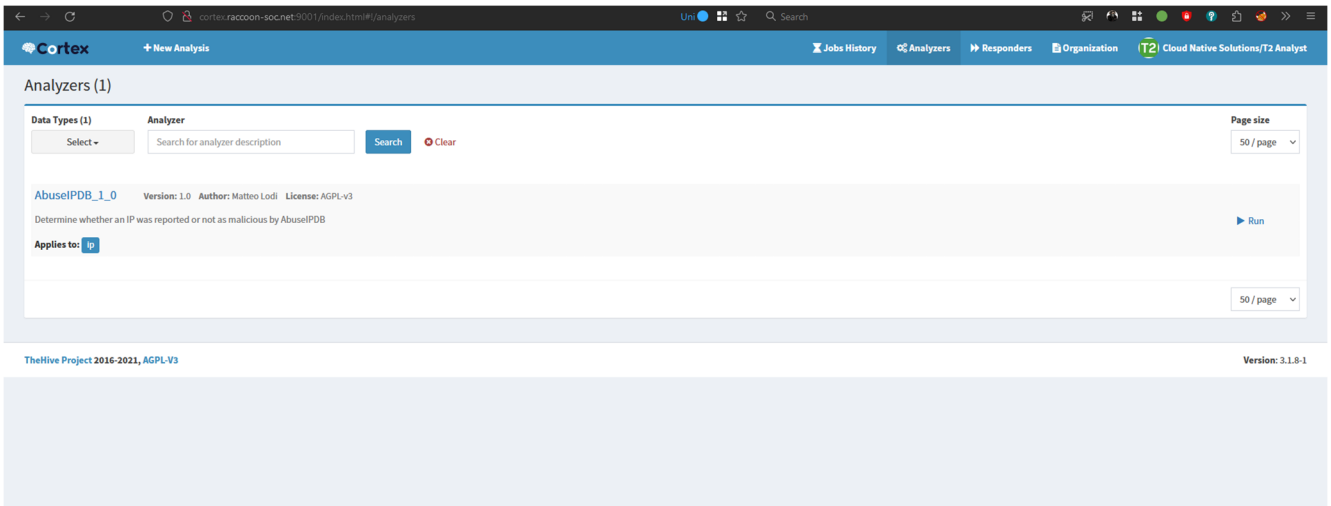

Рисунок 2.3.5 – Активований аналізатор AbuseIPDB_1_0 в Cortex

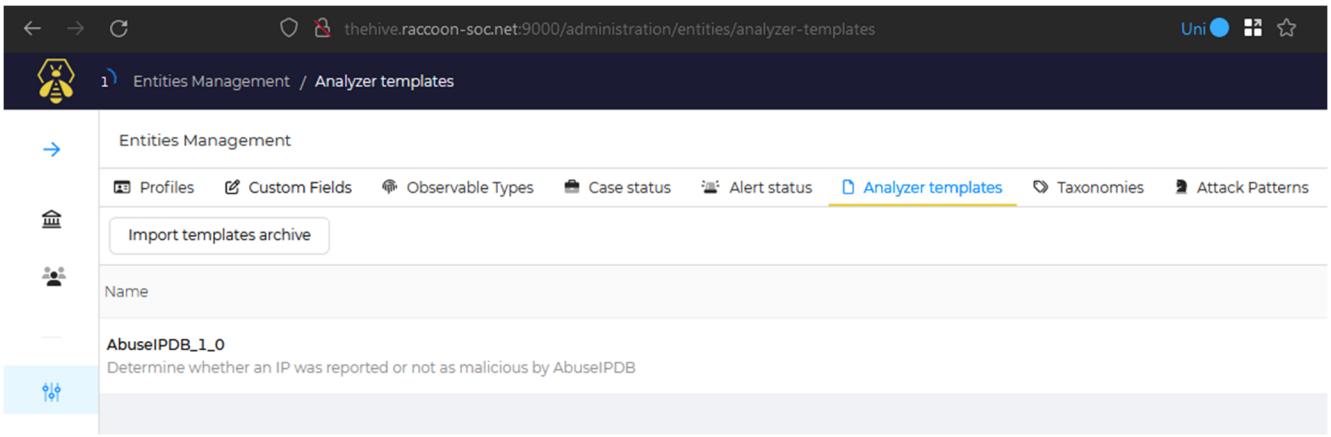

Рисунок 2.3.6 – Активований аналізатор AbuseIPDB_1_0 в TheHive

Цей аналізатор перевіряє репутацію тієї чи іншої IP адреси за допомогою сервісу AbuseIPDB. Перевіримо роботу цього аналізатору, попередньо вказавши в його налаштуваннях токен, через який він буде працювати з сервісом AbuseIPDB. Для його запуску через TheHive знадобиться створити тестовий кейс, додати до нього IP-адресу в якості спостережуваного значення (observable) та виконати над нею аналізатор.

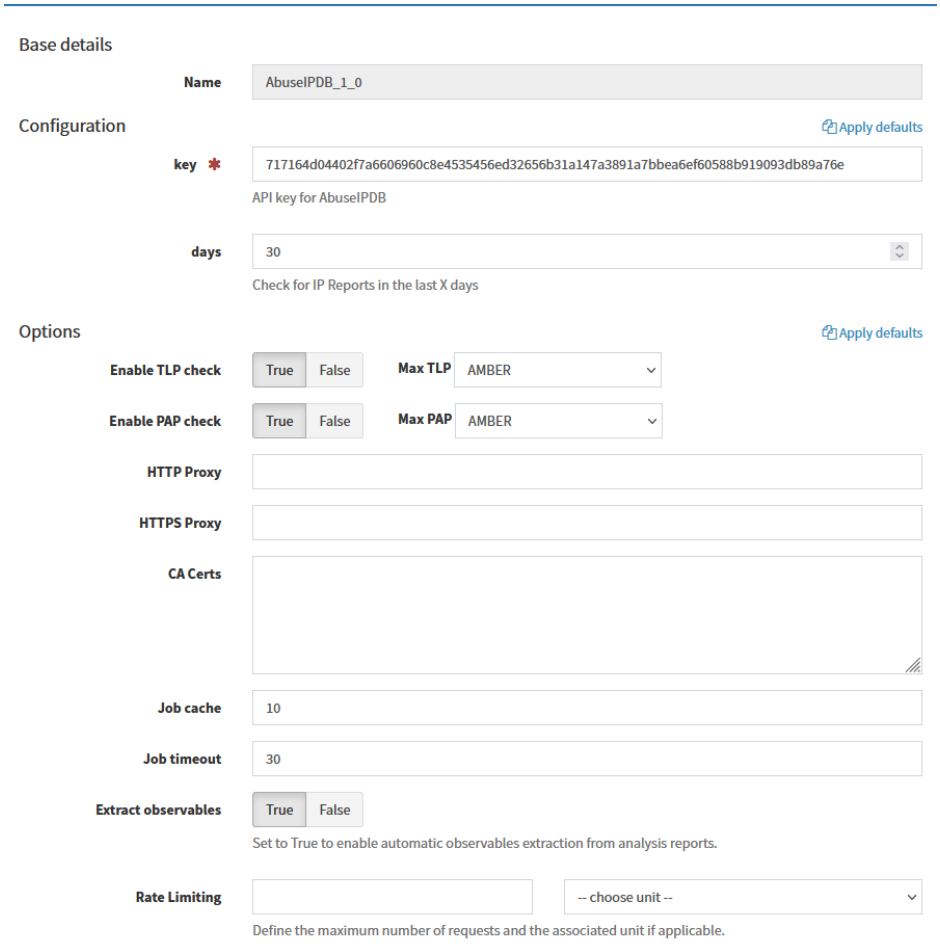

Рисунок 2.3.7 – Налаштування аналізатору AbuseIPDB

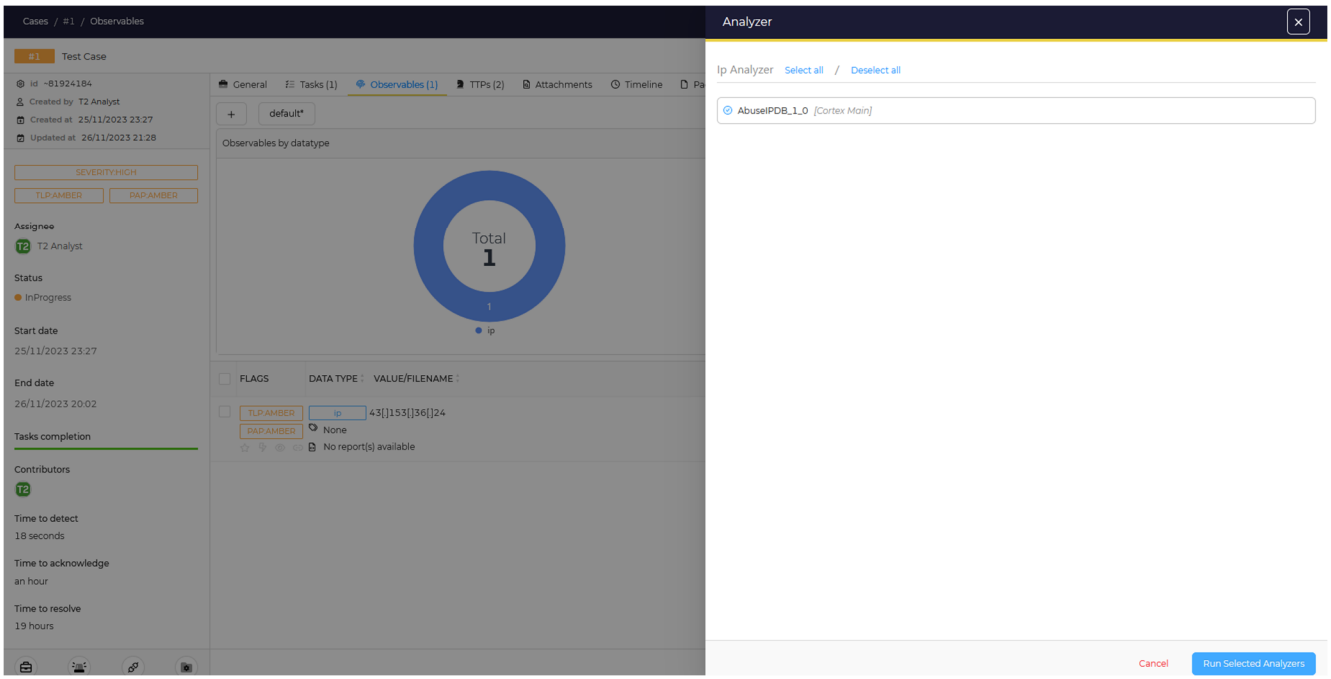

Рисунок 2.3.8 – Запуск аналізатора на зловмисному IP з тесового кейсу

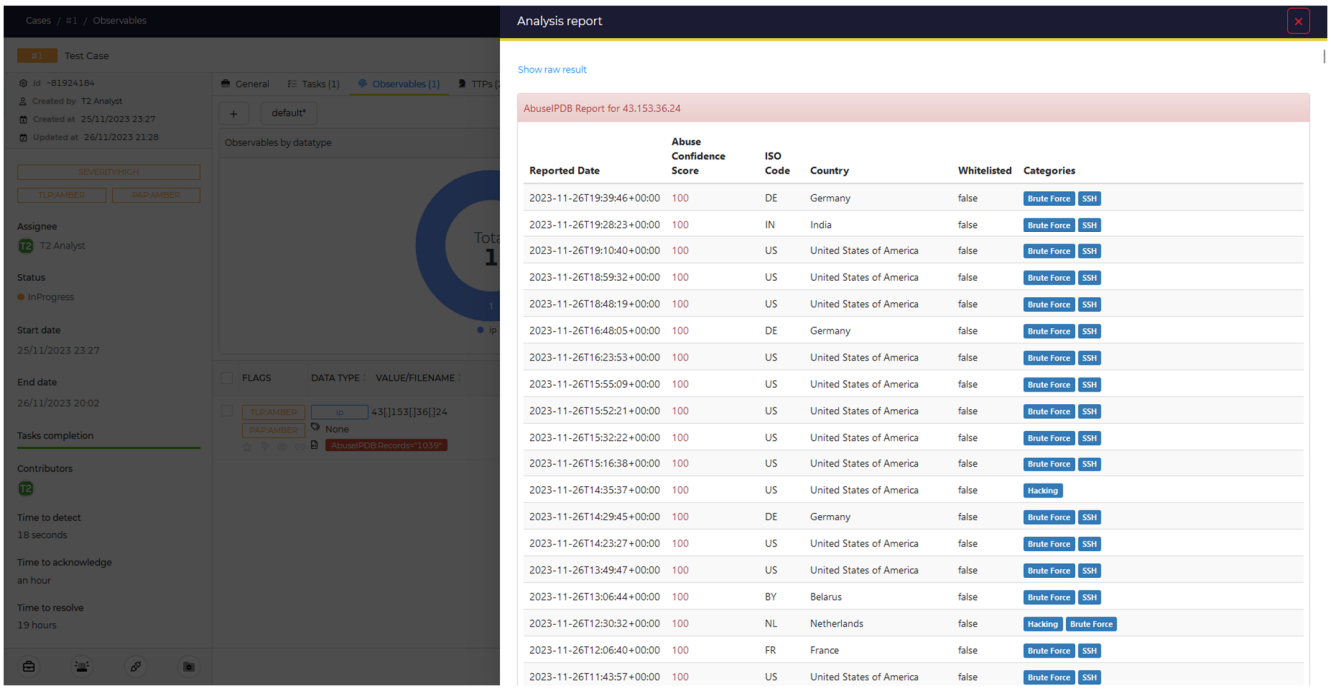

Рисунок 2.3.9 – Репутація IP-адреси, яку повернув аналізатор AbuseIPDB_1_0

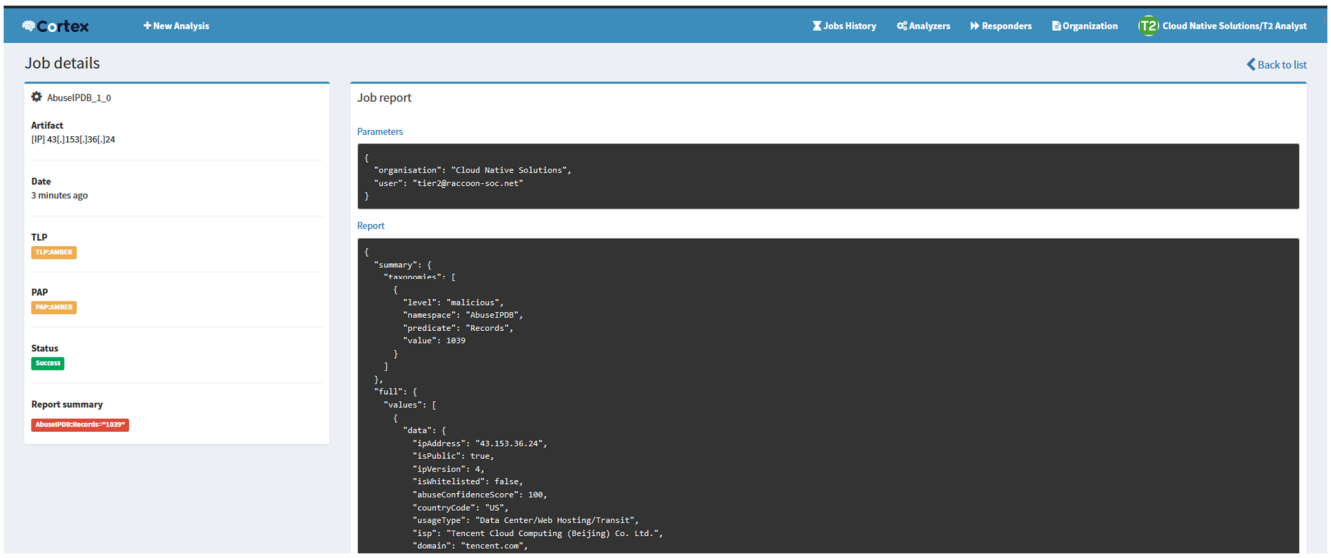

Рисунок 2.3.10 – Подробиці про запуск аналізатору AbuseIPDB_1_0 в Cortex

Рисунки 2.3.8 - 2.3.10 демонструють наявність повної зв’язності між TheHive та Cortex, а також вказують на те, що аналізована адреса дійсно зловмисна з 1039 звітами, більшість з яких вказують, що з цієї IP-адреси ведуться спроби SSH брутфорсу.

Наступним кроком буде перевірка виконання плейбуку, описаного в Shuffle системою SOAR (TheHive + Cortex).

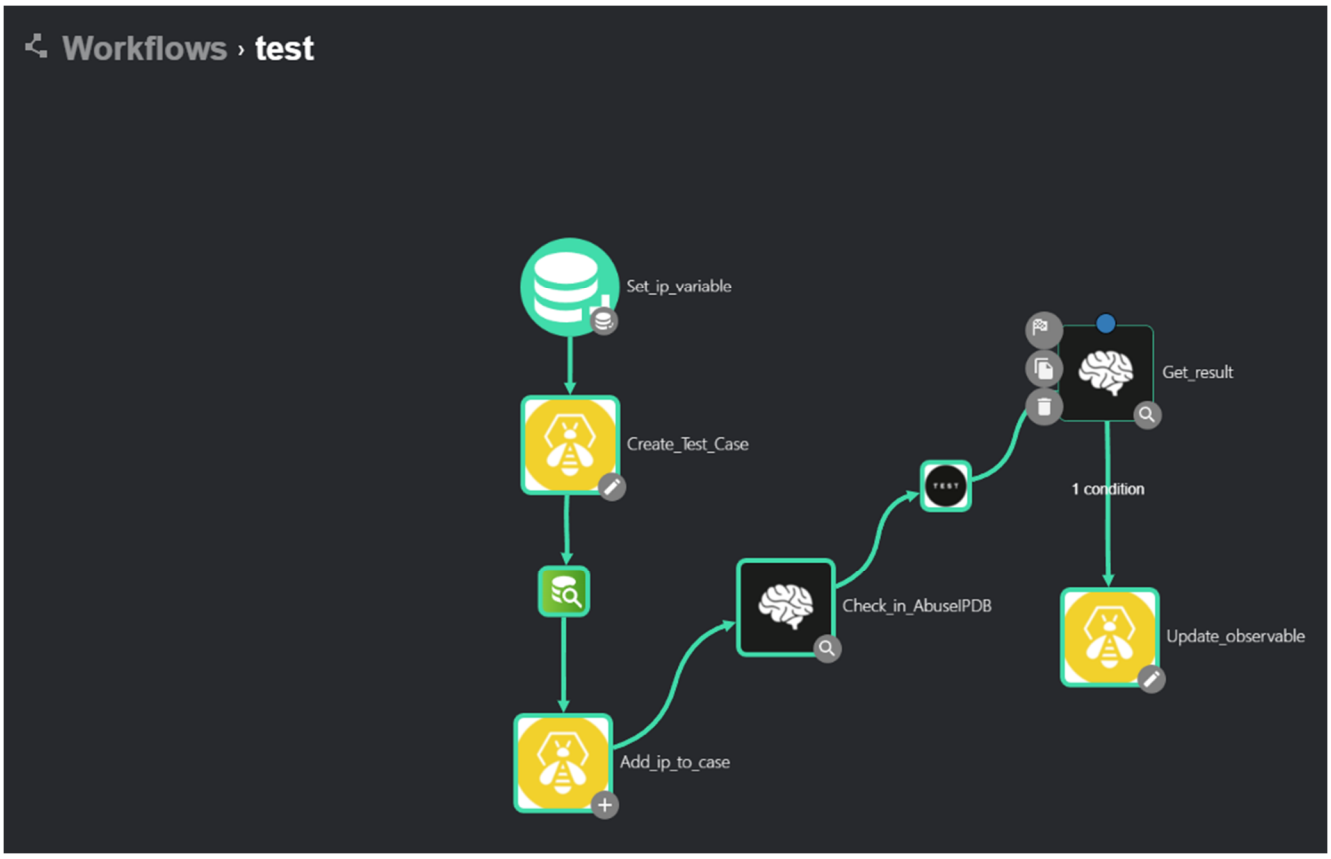

Рисунок 2.3.11 – Кроки тестового плейбуку

Плейбук на рис. 2.3.11 містить наступні кроки:

– Запис в змінну IP для перевірки.
– Створення кейсу в TheHive.
– Вичитка IP для перевірки зі змінної.
– Додавання IP до спостережуваних значеннь в створений кес.
– Перевірка IP адреси за допомогою Cortex аналізатору AbuseIPDB.
– Затримка в 30 секунд, щоб пройшла робота по перевірці IP в Cortex.
– Отримання результату перевірки.
– За умови, що перевірка повернула результати перейти на наступний крок.
– Додавання помітки IoC до спостережувного значення.

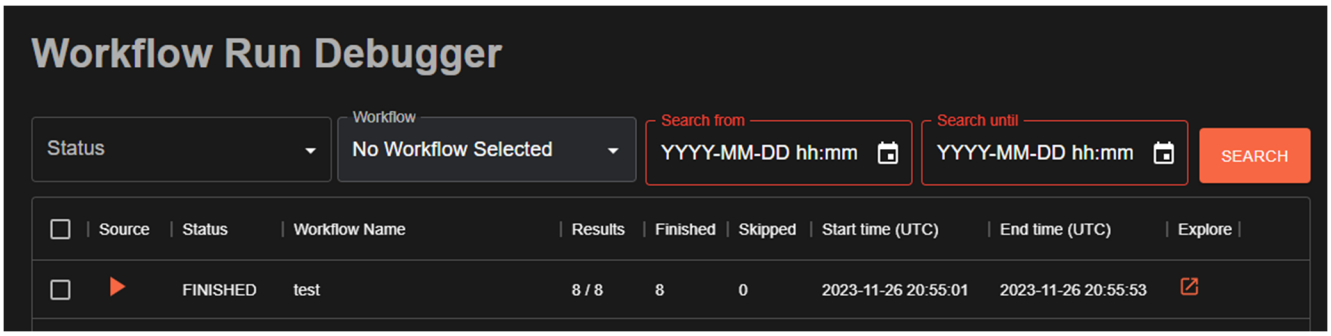

Рисунок 2.3.12 – Результат виконання плейбука в Shuffle

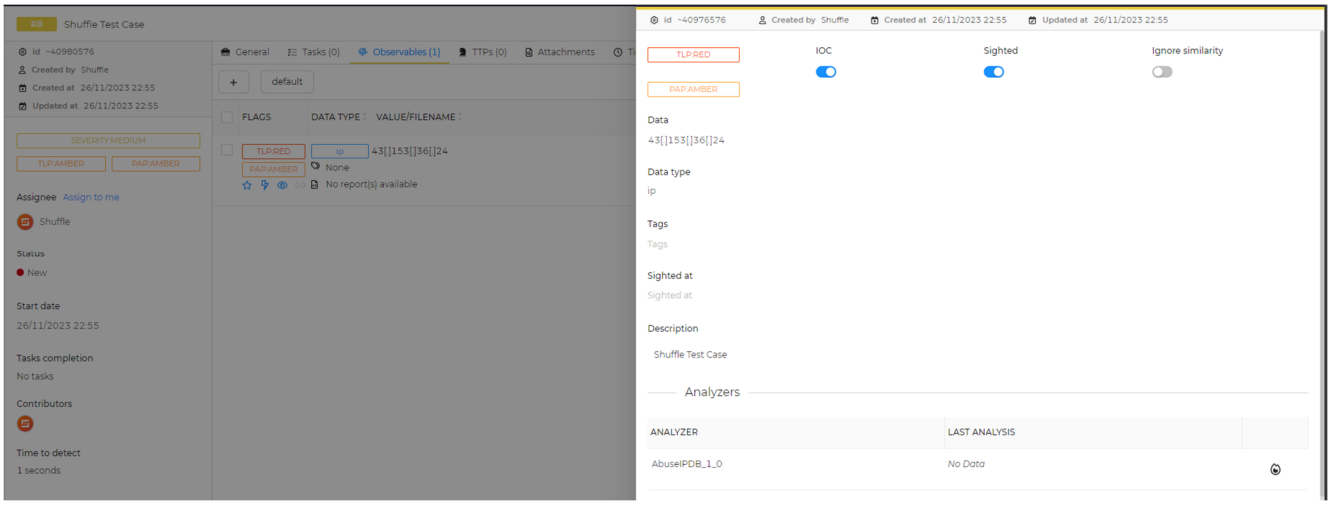

Рисунок 2.3.13 – Результат маніпуляцій плейбука в Cortex

Рисунок 2.3.14 – Результат маніпуляцій плейбука в TheHive

Рисунки 2.3.12. – 2.3.14 демонструють успішний запуск та виконання створеного плейбуку, що свідчіть про успішний зв’язок між всіма наявними компонентами SOC. Як простір для покращення можна виділити додавання можливості імпорту алертів з SIEM ELK та створення на їх основі сповіщень (алертів) в TheHive, які можуть бути підвищені до рівня інцидентів (кейсів) через плейбук Shuffle. Саме цю можливість, а також решту компонентів буде вбудовано в SOC в ході виконання магістерської роботи.

ВИСНОВКИ

На початку проходження переддипломної практики було обґрунтовано актуальність теми дослідження, а саме проєктування підсистему моніторингу подій безпеки SOC на базі безкоштовних відкритих рішень.

Також був проведений огляд та аналіз ринку сучасних безпекових FOSS рішень, що в свою чергу дало змогу обрати конкретні рішення SIEM та SOAR у якості бази майбутнього SOC.

Наступним кроком виступало проєктування самої інфраструктури SOC на базі FOSS рішень, створення віртуальної мережі в середовищі емуляції, налаштування мережевих пристроїв, серверів та самих безпекових FOSS рішень.

З метою перевірки взаємодії кожного з компонетів між собою було розроблено тестовий плейбук в середовищі Shuffle. Це дозволило наочно продемонструвати, яким чином та за допомогою яких процесів та механізмів SIEM та SOAR дозволяють аналітикам SOC реагувати на події безпеки в ручному та автоматизованому режимах.

Як результат – спроєктовано та налаштовано робочу платформу SIEM + SOAR, яка при подальшому вдосконалені у ході магістерської роботи перейде в стан повноцінного та працюючого SOC.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корпоративна безпека : Практ. посіб. Київ : Консалтинг. компанія Сідкон, 2018. 276 с.
2. SOC adoption: three scenarios to a better security posture : Електронна книжка. Infopulse, 2020. 40 с.
3. Bejtlich R. The practice of network security monitoring: understanding incident detection and response. No Starch Press, 2013. 376 с.
4. Thomas A. Security operations center - analyst guide: SIEM technology, use cases and practices. Arun E Thomas, 2017. 206 с.
5. Murdoch D. Blue team handbook: SOC, SIEM, and threat hunting (V1.02): a condensed guide for the security operations team and threat hunter. Independently published, 2019. 258 с.
6. SIEM Home Lab Series (Part 1) [Електронний ресурс] // Jared Bloomberg. – 2020. – Режим доступу до ресурсу: https://unicornsec.com/home/siem-home-lab-series-part-1.
7. Valente Labs. Installing Elasticsearch Stack [Part 1] [Електронний ресурс] / Valente Labs // Medium. – 2021. – Режим доступу до ресурсу: https://valentelabs.medium.com/installing-elasticsearch-stack-part-1-9dd9b8912890.
8. TheHive: Step-by-Step guide [Електронний ресурс] // StrangeBee. – 2022. – Режим доступу до ресурсу: https://docs.strangebee.com/thehive/setup/installation/step-by-step-guide/.
9. Cortex: Installation & configuration guides [Електронний ресурс] // StrangeBee. – 2022. – Режим доступу до ресурсу: https://docs.strangebee.com/cortex/installation-and-configuration/.
10. Arfath M. SOC implementation with TheHive, Cortex & Elasticsearch [Електронний ресурс] / Mohomed Arfath // devgenius.io. – 2022. – Режим доступу до ресурсу: https://blog.devgenius.io/soc-implementation-with-thehive-cortex-elasticsearch-672e89219f0c.