Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів”

Виконав здобувач вищої освіти
II курсу, групи КІ-25М
ОПП «Комп’ютерна інженерія»
спеціальності F7 «Комп’ютерна інженерія»
_____ Рескаленко Я.Ю.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Буравченко К.О.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Рескаленко Ярослав Юрійович
Сертифікат: 18DF2C9B93CEA20404000000AB1A5200B9D27A06
Дійсний з 24.11.2026 16:57:28 по 24.11.2027 16:57:27

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _F_____*"Інформаційні технології"*_____

Спеціальність F7 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Рескаленку Ярославу Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів*

2. Керівник роботи *Буравченко Костянтин Олегович, кандидат технічних наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 41-13 від 17.07.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна.*
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*7. Висновки.*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

7. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
7.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Рескаленко Я.Ю. Дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів. F7 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Метою розробки є дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Об'єктом дослідження є процес мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Предметом дослідження є методи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп’ютерна інженерія, управління інформаційною безпекою (SOC), модуль автоматизованого реагування, оркестрація інцидентів

ABSTRACT

Reskalenko Ya.Yu. Research and Software Implementation of a Security Operations Center (SOC) System with Automated Incident Response and Orchestration Modules. F7 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This final qualification thesis for the second (master's) level of higher education presents the development of software designed for a Security Operations Center (SOC) system, featuring modules for automated incident response and orchestration.

The objective of this development is the research and software implementation of a Security Operations Center (SOC) system with automated incident response and orchestration modules.

The object of the research is the process of operating a Security Operations Center (SOC) with automated incident response and orchestration modules.

The subject of the research consists of the methods employed within a Security Operations Center (SOC) with automated incident response and orchestration modules.

The research methodology draws upon the theory of computer network design, information security theory, mathematical statistics, and software development practices.

The outcome of the work is the software implementation of a Security Operations Center (SOC) system with automated incident response and orchestration modules.

During the development of the software model, an analysis of existing hardware and software tools was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed. Instructions for using the software tools are provided.

The program can be used on PCs running the Windows 10/11 operating system.

The program was developed in the Python environment.

Keywords: computer engineering, information security management (SOC), automated response module, incident orchestration

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	83	–	
2	А4	Робоча програма	49	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 10
ВСТУП..... 11
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 14
1.1 Призначення системи..... 14
1.2 Область застосування 15
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 17
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 17
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 24
2.3 Розгорнута постановка завдання 24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 26
3.1 Опис функціонування системи 26
3.2 Розробка структурної схеми..... 28
3.3 Розробка функціональної схеми 39
3.4 Розробка діаграми процесів 42
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 44
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 44
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 59
6 НАУКОВА НОВИЗНА 64

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 65
7.1 Вступ 65
7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером 66
7.3 Розробка заходів з умов поліпшення охорони праці 67
7.4 Пожежна безпека 68
7.5 Розрахункова частина 71
8 ОСНОВНІ ВИСНОВКИ 74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ГБЗ	–	глобальна база знань
ДФ	–	дестабілізуючі фактори
ІС	–	інформаційна система
КФ	–	керуючі фактори
ЛОМ	–	локальна обчислювальна мережа
НКК	–	нечіткі когнітивні карти
ММЕ	–	міжмережний екран
ПФ	–	проміжні фактори
СЗДДДТ	–	систем запобігання деструктивним діям дестабілізуючого трафіку
ЦФ	–	цільові фактори
SSDT	–	таблиця дескрипторів системних сервісів

ВСТУП

Актуальність теми. Центр операцій безпеки (SOC) – це термін, який ви часто чутимете, оскільки він є важливою частиною будь-якої організації, яка використовує різні технології (наприклад, хмарні системи, пристрої віддаленого доступу, мобільні пристрої тощо). SOC відповідає за відстеження великої кількості інформації з різних джерел та звітування про її стан вищому керівництву.

У даній роботі надасться огляд інструментів, доступних для функціонування SOC, персоналу, необхідного для роботи SOC, та щоденних стандартних операційних процедур (СОП) типового SOC. Окреслимо кілька найкращих практик, які можуть допомогти організаціям швидше виявляти потенційні загрози, ефективніше реагувати на них та постійно вдосконалювати свої можливості реагування на події безпеки.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– Дослідження системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– Програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Об’єктом дослідження є процес мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Предметом дослідження є методи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Методи дослідження базуються на методах теорії побудови комп’ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– Розроблено вітчизняний продукт мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Якщо ви коли-небудь задавалися питанням, чому компанії інвестують значні кошти в команди SOC, ось проста відповідь: ситуація з загрозами не сповільнюється. Зловмисники діють швидко. SOC допомагає організаціям залишатися на крок попереду. Ось що може запропонувати сильна SOC:

– Проактивне зменшення загроз: Зловмисники більше не проникають силою, а ховаються або рухаються непомітно та затримуються. Завдання SOC полягає у ранньому виявленні та усуненні таких загроз, а також у зупинці їх, перш ніж вони зможуть завдати значної шкоди.

– Швидке реагування на інциденти: У разі порушення кожна мить важлива. SOC може мінімізувати час між моментом, коли «подія сталася», і моментом, коли «її було локалізовано».

– Цілодобовий моніторинг: Атаки відбуваються 24 години на добу, 7 днів на тиждень. SOC постійно контролює системи, включаючи мережевий трафік, журнали на всіх серверах, кінцевих точках та хмарних середовищах.

– Судово-медична експертиза: Після інциденту SOC надасть детальну інформацію про подію, що сталося, як це було здійснено та як запобігти повторенню.

– Управління вразливостями: Хоча SOC може надавати реактивні ресурси, він також може виявляти будь-які наявні вразливості та допомагати організації у вирішенні цих вразливостей.

– Розвідка про загрози: Центр захисту даних (SOC) слідкує за зловмисниками, їхніми методами та схемами атак, щоб вони не були здивовані в майбутньому.

Підсумовуючи: SOC – це цифровий захисник сучасного бізнесу. Без нього ризик компрометації зростає в геометричній прогресії.

1.2 Область застосування

Для забезпечення ефективної роботи Центру операцій безпеки (ЦОБ), ЦОБ повинен мати найкращі доступні технології для забезпечення цих операцій. Основні технології, що забезпечують роботу ЦОБ, включають:

1. SIEM – Управління інформацією та подіями безпеки. SIEM – це основна технологія, яка збирає журнали з джерел у вашому середовищі. SIEM також дозволяє аналітикам безпеки виявляти будь-яку незвичайну або підозрілу поведінку з цих журналів.

2. IDS/IPS – Системи виявлення та запобігання вторгненням. Ці технології допомагають аналітику безпеки, збираючи та моніторячи вхідний/вихідний мережевий трафік для виявлення підозрілої активності. IDS надає сповіщення безпеки про підозрілу активність, тоді як IPS забезпечує реагування в режимі реального часу, блокуючи атаку, виявлену IDS.

3. Виявлення кінцевої точки та реагування – EDR. Завдяки своїй доступності кінцеві точки стають легкою мішенню для кіберзлочинців. Інструменти EDR дозволяють командам безпеки контролювати кінцеві точки на предмет законного використання та виявляти аномальну або нетипову поведінку, а також ізолювати пристрої, що перебувають у карантині, та мінімізувати їхню шкоду, перш ніж вони завдадуть подальшої шкоди.

4. Платформи розвідки загроз. Платформи розвідки загроз – це сукупність даних про загрози, зібраних з різних джерел, включаючи, але не обмежуючись, стрічки даних про загрози, публічні індикатори компрометації та галузеві звіти. Аналітики безпеки використовують ці дані, щоб допомогти передбачити, де відбуватимуться майбутні атаки.

5. Автоматизація та оркестрація – SOAR. Автоматизація та реагування на оркестрацію безпеки (SOAR) дозволяє аналітикам безпеки автоматизувати ручні процеси, пов'язані з повторюваними завданнями, що виконуються за допомогою кількох інструментів безпеки, та створювати належні робочі процеси для аналітиків безпеки, щоб усі технології безпеки могли працювати разом

узгоджено одна з одною, щоб аналітики витрачали більше часу на проведення розслідувань, а не на натискання кнопок.

6. Система управління справами. Система управління справами (CMS) слугує «робочим простором» для центру операцій безпеки. CMS зберігатиме всю інформацію про інциденти безпеки, зібрані докази, записані нотатки та оновлення статусу, щоб персонал SOC не втратив жодних критичних доказів інциденту безпеки.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Зайдіть сьогодні до будь-якого корпоративного SOC, і ви знайдете приблизно таку саму структуру. SIEM у центрі, EDR, що постачає йому дані з кінцевих точок, SOAR, що виконує сценарії, і черга сповіщень, яку ніхто не встигає належним чином дослідити. Технічно стек працює, але запитайте аналітиків, які дивляться на нього, чи довіряють вони тому, що до них доходить, і відповідь рідко буде ствердною.

Більшість команд SOC більше не обмежуються збором телеметрії. Вони намагаються вирішити, що заслуговує на увагу людини. Опитування SANS Detection and Response Survey 2025 року показало, що 73% організацій тепер називають хибнопозитивні результати своєю головною проблемою виявлення, а аналіз The Hacker News показує, що аналітики втрачають до 30% свого часу на їх переслідування. Постачальники відреагували агресивно. «На базі штучного інтелекту» тепер означає радикально різні речі залежно від того, кого ви запитаєте, а рішення, які колись зводилися до витрат, тепер залежать від того, наскільки добре платформа розуміє середовище, яке вона захищає.

Протягом більшої частини останнього десятиліття відповіддю на шумний SOC було більше інструментів, але команди перестали запитувати, як швидше обробляти більше сповіщень, і почали запитувати, чому їхні інструменти не можуть сказати їм, які сповіщення важливі в першу чергу.

– Проблема обсягу сповіщень, яку традиційні інструменти SOC не можуть вирішити: навіть після застосування автоматизації обсяг сповіщень, що надходять до кожного аналітика, все ще перевищує те, що одна людина реально може дослідити за один день. SIEM агрегуються, SOAR оркеструються, а EDR

сигналізують про події кінцевих точок. Додавання додаткових інформаційних панелей на додаток до цієї прогалини не закриває її.

– Прихована ціна ручного сортування та вигорання аналітиків: Видима ціна – це години, витрачені аналітиками на сповіщення, які виявляються беззагрозними. Прихована ціна – це втома від сповіщень, яка накопичується під ними, притупляючи судження, на які покладаються аналітики, коли реальний інцидент нарешті з'являється в черзі.

– Чому операційний контекст визначає якість розслідування: Агент штучного інтелекту, який закриває сповіщення за 30 секунд, не розуміючи середовища, просто швидше генерує погані висновки. Достовірні вердикти від ненадійних відрізняє операційний контекст, зокрема, кому належить уражений актив, який бізнес-процес він підтримує та що аналітик насправді зробив би, якби мав час провести належне розслідування.

– Що відрізняє платформу SOC на основі штучного інтелекту від автоматизації, вбудованої в SIEM: Автоматизація, вбудована в SIEM, передбачає, що безпека є детермінованою, тому вона дотримується статичних сценаріїв, доки середовище не зміниться та робочий процес не зламається. Платформа SOC на основі штучного інтелекту досліджує та адаптується, ґрунтуючи кожен вердикт на організаційному контексті, а не на правилі, написаному 18 місяців тому.

Mate Security

Mate Security – це агентна SOC-платформа, побудована на основі графу контексту безпеки (Security Context Graph), моделі, що постійно розвивається, як насправді функціонує організація, включаючи її стандартні операційні процедури (SOP), відносини власності, операційні залежності та попередні рішення щодо розслідувань. Платформа виконує виявлення, сортування, розслідування та реагування як один безперервний цикл, причому кожне сповіщення повертає граф до нових міркувань.

Видатні можливості: Графік будується протягом 24 годин та обґрунтовує розслідування в операційному середовищі, а не покладається на загальні базові

припущення. Контрольоване реагування виконує дії відповідно до внутрішніх процедур, водночас дозволяючи аналітикам контролювати прийняття рішень з високим рівнем впливу. Серед корпоративних клієнтів – Bridgewater Associates, Lead Bank, AlphaSense та Merlin Entertainments, а опубліковані показники інформаційної панелі Mate показують, що одне розгортання скоротило MTTR на 93% протягом 5 місяців.

Де він перевершує інших: Корпоративні SOC, що працюють з великими обсягами сповіщень та втомою аналітиків, де глибина кожного вердикту та обґрунтування його мають більше значення, ніж додавання додаткових правил до вже й так шумного стеку виявлення.

Microsoft Sentinel

Microsoft Sentinel перетворився з хмарної SIEM на ширшу платформу операцій безпеки, що поєднує єдине озеро даних, єдину видимість розслідувань, SOAR, UEBA та аналітику загроз на порталі Microsoft Defender. Security Copilot вбудований для робочих процесів розслідування природною мовою, а розміщений сервер Model Context Protocol забезпечує на платформі інструменти, готові до роботи з агентами.

Видатні можливості: Бібліотека з понад 350 власних конекторів даних забезпечує широке покриття в багатохмарних та багатоплатформних середовищах, а міграція SIEM за допомогою штучного інтелекту перетворює виявлення Splunk та QRadar на власні правила Sentinel зі зменшенням ручних зусиль. Безкоштовна підтримка міграції також доступна через програму Microsoft Cloud Accelerate Factory.

Де він перевершує інших: Середовища, що вперше розроблені Microsoft, стандартизовані на Defender XDR, Entra та Purview, де нативна інтеграція між ідентифікаторами, кінцевими точками та хмарною телеметрією є найглибшою.

Splunk Enterprise Security

Тепер, як частина Cisco, Splunk Enterprise Security перепозиціонував себе як єдина платформа для виявлення, розслідування та реагування на загрози (TDIR), доступна у версіях Essentials та Premier. Mission Control консолідує

робочі процеси аналітиків, тоді як Splunk AI Assistant in Security та зростаючий набір агентних можливостей штучного інтелекту розширюють платформу ще більше на операції SOC за допомогою штучного інтелекту.

Видатні можливості: Система оповіщення на основі ризиків зменшує шум оповіщень, об'єднуючи пов'язані події в пріоритетні інциденти, а не спрацьовуючи за кожним сигналом окремо, тоді як розвідка загроз Cisco Talos інтегрована безпосередньо в робочі процеси розслідування. Дослідження IDC, на яке посилається Splunk, показує 304% рентабельність інвестицій та на 64% швидше виявлення загроз для організацій, що використовують єдину платформу TDIR.

Де він перевершує інших: Великі підприємства зі зрілими командами SOC та глибоким досвідом Splunk, де основними операційними вимогами залишаються розширена логіка кореляції, налаштування та можливість обробки дуже великих обсягів телеметрії.

CrowdStrike Falcon

CrowdStrike Falcon – це платформа для операцій безпеки на базі штучного інтелекту, яка поєднує безпеку кінцевих точок, XDR, розвідку загроз, захист ідентифікаційних даних, хмарну безпеку, SOAR та можливості SIEM наступного покоління на основі платформи Falcon та архітектури даних LogScale без індексів. Charlotte AI та Charlotte Agentic SOAR розширюють платформу розслідуваннями за допомогою штучного інтелекту, автоматизацією робочих процесів та оркестрацією агентів без коду для сучасних операцій SOC.

Видатні можливості: вбудоване використання телеметрії Microsoft Defender для кінцевих точок, федеративний пошук у розподілених сховищах даних і агент перетворення запитів, який перетворює пошукові запити Splunk на мову запитів CrowdStrike, зменшують труднощі для організацій, які мігрують зі застарілих середовищ SIEM. Falcon Fusion SOAR та Charlotte AI допомагають аналітикам автоматизувати розслідування та робочі процеси реагування на телеметрію кінцевих точок, ідентифікаційних даних та хмари.

Де це перевершує інших: Організації, які вже стандартизували

CrowdStrike, де тісна інтеграція між робочими процесами кінцевих точок, ідентифікації, хмари та розвідки загроз може оптимізувати операції SOC та зменшити складність розслідувань.

Palo Alto Networks Cortex XSIAM

Cortex XSIAM – це консолідована платформа для операцій безпеки Palo Alto Networks, яка об'єднує робочі процеси SIEM, SOAR, виявлення кінцевих точок, мережі та хмарних технологій на Cortex Extended Data Lake. Cortex AgentiX нашаровує можливості агентного штучного інтелекту на платформу, розширюючи основу автоматизації, спочатку створену Cortex XSOAR, за допомогою попередньо створених агентів для розвідки загроз, розслідування електронної пошти, криміналістики кінцевих точок та мережевого реагування.

Видатні можливості: AgentiX побудовано на даних, отриманих з понад 1,2 мільярда реальних виконань сценаріїв, та включає понад 1000 попередньо створених інтеграцій, а також власну підтримку протоколу контексту моделі (MODE). Palo Alto Networks повідомляє про скорочення MTTR до 98% та на 75% менше ручної роботи в клієнтських середовищах, згідно з внутрішніми бенчмарками.

Де він досягає успіху: Підприємства, які прагнуть консолідації платформ, зокрема клієнти Palo Alto Networks, які прагнуть об'єднати SOC, кінцеві точки, мережу та хмарну телеметрію в рамках єдиної екосистеми.

ServiceNow Security Operations

ServiceNow Security Operations – це платформа для операцій безпеки та SOAR, керована робочими процесами, побудована на платформі Now, яка об'єднує реагування на інциденти безпеки, реагування на вразливості та розвідку загроз на основі ширшої бази ITSM та CMDB ServiceNow. Now Assist додає можливості агентного штучного інтелекту для підсумовування інцидентів, планування вирішення за допомогою наборів завдань, статей знань та минулих подібних інцидентів, а також звітування про передачу зміни природною мовою.

Видатні можливості: Вбудована інтеграція CMDB фіксує інциденти в контексті активів, користувачів та послуг, тоді як інтеграція Microsoft Security

Copilot забезпечує двонаправлену співпрацю між ServiceNow SIR та даними безпеки Microsoft на основі штучного інтелекту. AI Control Tower розширює управління агентами, моделями та ідентифікаторами штучного інтелекту за допомогою журналів аудиту, метрик впровадження, моніторингу виконання, контролю ризиків та відстеження цінності.

Де це перевершує: Підприємства вже стандартизували ServiceNow для IT та бізнес-робочих процесів, де консолідація реагування на інциденти безпеки в одній системі записів може зменшити передачу обов'язків між командами SOC, IT та відповідності.

Wiz

Wiz розширилася від платформи управління хмарною безпекою до ширшої екосистеми CNAPP та виявлення в хмарі, поєднуючи DSPM, управління вразливостями, аналіз шляхів атак та виявлення в хмарі в рамках єдиної графової архітектури. Wiz Defend розширює платформу до хмарних робочих процесів виявлення та реагування з розслідуванням за допомогою штучного інтелекту, кореляцією загроз та багатохмарною видимістю.

Видатні можливості: Wiz Security Graph співвідносить ідентифікаційні дані, робочі навантаження, секрети, вразливості, мережевий ризик та токсичні комбінації в уніфіковані шляхи атак, які надають пріоритет реальним бізнес-ризикам, а не окремим результатам. Вбудовані інтеграції між AWS, Azure, Google Cloud, Kubernetes та інструментами розробника забезпечують командам безпеки широку видимість з мінімальними операційними витратами.

Де він перевершує інших: хмарні підприємства, що працюють у великих багатохмарних середовищах, особливо організації, що надають пріоритет управлінню впливом, аналізу горизонтального руху та видимості шляхів хмарних атак поряд з традиційними розслідуваннями SOC.

Microsoft Defender XDR

Microsoft Defender XDR – це інтегрована розширена платформа Microsoft для виявлення та реагування, яка об’єднує телеметрію кінцевих точок, ідентифікаційних даних, електронної пошти, хмарних програм та

інфраструктури в Defender for Endpoint, Defender for Identity, Defender for Office 365 та Defender for Cloud Apps. Платформа тісно інтегрується з Microsoft Sentinel для ширших операцій SIEM та SOAR, тоді як Security Copilot додає робочі процеси розслідування на основі штучного інтелекту з пошуком природної мови, автоматичним підсумовуванням, рекомендаціями щодо керованого реагування та допомогою в сортуванні.

Видатні можливості: Міждоменна кореляція автоматично пов'язує сигнали кінцевих точок, ідентифікаційних даних, електронної пошти та хмари в єдині інциденти, а глибока інтеграція з Entra, Purview та ширшою екосистемою безпеки Microsoft надає аналітикам чітку контекстуальну видимість у різних корпоративних середовищах. Керування ризиками, переривання атак та вбудована автоматизація допомагають зменшити витрати на ручне розслідування у масштабних операціях SOC.

Де він перевершує інших: Корпорації, орієнтовані на Microsoft, які шукають тісно інтегровані операції XDR та SOC в інфраструктурі кінцевих точок, ідентифікації, SaaS та хмари, не покладаючись суттєво на розрізнені сторонні інструменти.

NetWitness: Сильний партнер для сучасних державних операційних компаній

NetWitness має поважну репутацію бренду в галузі кібербезпеки не без підстав. Оскільки зловмисники стають все більш витонченими, NetWitness постійно вдосконалює свої функції, щоб допомогти командам SOC залишатися на крок попереду.

NetWitness вирізняється своєю здатністю інтегрувати видимість усього середовища – журналів, мережевого трафіку, кінцевих точок, хмари, а також пристроїв Інтернету речей. Ця узгоджена модель даних, разом з автоматизованою аналітикою та функціями SOAR, надає аналітикам комплексне уявлення, а не розрізнені висновки.

Більшість постачальників продовжують збирати рішення різними способами. NetWitness SOC пропонує цілісну платформу, яку організації можуть

впроваджувати як SaaS, локально або через керований сервіс. Незалежно від того, чи ви керуєте вже існуючим SOC, чи створюєте його з нуля, NetWitness надає аналітичні дані, гнучкість та знання, необхідні для запобігання вторгненням, що дозволяє вам спокійніше проводити ніч.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Мова програмування Python є високоефективним інструментом для розробки сценаріїв автоматизованого реагування (SOAR), збору телеметрії та інтеграції компонентів у сучасних центрах управління інформаційною безпекою (SOC). Завдяки високорівневим структурам даних, гнучкій об'єктно-орієнтованій парадигмі та виразному синтаксису вона оптимально підходить для швидкого прототипування алгоритмів кореляції подій та оркестрації інцидентів кібербезпеки.

Офіційні дистрибутиви та розширювані компоненти інтерпретатора Python доступні на веб-сайті, що забезпечує доступ до потужної екосистеми бібліотек для аналізу логів та взаємодії із захищеними API. Завдяки відкритій архітектурі інтерпретатор легко доповнюється низькорівневими функціями й типами даних, написаними на C або C++, що дозволяє оптимізувати обробку інтенсивних потоків мережевого трафіку та реалізовувати високошвидкісні модулі автоматизованого реагування на інциденти в середовищі SOC.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Найкращі практики для сильного SOC

Високопродуктивний SOC не створюється за одну ніч. Ось практики, які забезпечують його ефективність та готовність до будь-яких ситуацій:

– Управління життєвим циклом інцидентів: Чіткі кроки для виявлення, аналізу, стримування та відновлення після загроз.

– Полювання на загрози: активний пошук прихованих загроз замість очікування сповіщень.

– Визначені схеми дій: попередньо розроблені відповіді, які швидко та послідовно допомагають аналітикам вживати правильних заходів.

– Регулярне навчання: Загрози швидко змінюються. Команди SOC залишаються активними завдяки постійному навчанню та реальним навчанням.

– Постійне вдосконалення: Інструменти, правила та процеси необхідно часто оновлювати, щоб випереджати нові загрози.

Проблеми SOC та як команди їх вирішують

Керувати SOC непросто. Ось найбільші труднощі, з якими стикається більшість команд:

– Еволюція загроз: Зловмисники постійно експериментують. SOC покладаються на розвідку та аналітику загроз для відстеження нових тактик.

– Нестача кваліфікованих фахівців: важко знайти фахівців у сфері безпеки. Організації долають цей розрив за допомогою кращого навчання, автоматизації та керованих послуг.

– Перевантаження даними: Величезні обсяги журналів можуть перевантажити аналітиків. Системи SIEM та автоматизоване сортування допомагають зменшити шум.

– Складні вимоги до дотримання вимог: Від GDPR до галузевих правил,

SOC допомагають підтримувати дотримання вимог шляхом моніторингу, звітності та забезпечення дотримання політик.

– Проблеми інтеграції інструментів: Багато організацій накопичують інструменти, які не взаємодіють один з одним. Безпека SOAR допомагає уніфікувати стек.

Переваги впровадження SOC

Коли SOC працює добре, вся організація відчуває на собі вплив:

– Нижчий ризик кібербезпеки.

– Швидше та більш скоординоване реагування на інциденти.

– Краща видимість у мережах та кінцевих точках.

– Простіша звітність про дотримання вимог.

– Довгострокова економія коштів завдяки уникненню масштабних порушень.

SOC – це більше, ніж функція безпеки, це стратегічна інвестиція в операційну стабільність.

Кібербезпека швидко змінюється, і компанії з управління охороною здоров'я розвиваються разом з нею. Ось тенденції, що формують майбутнє:

– Штучний інтелект та машинне навчання.

– Автоматизоване виявлення, розумніша аналітика та швидші розслідування.

– Хмарні моделі SOC.

– Безпека, що масштабується з гібридними та багатохмарними середовищами.

– Обмін інформацією про загрози.

– Більше співпраці між галузями для раннього виявлення закономірностей.

– Безпека Інтернету речей.

– Більше пристроїв означає більше поверхонь для атак. SOC тепер контролюють усе: від ноутбуків до промислових датчиків.

3.2 Розробка структурної схеми

Центр операцій безпеки – це централізована функція, яка постійно контролює, виявляє, розслідує та реагує на кіберзагрози в середовищі організації. За однією з оцінок, світовий ринок SOC оцінювався в 42,85 мільярда доларів США у 2024 році та, за прогнозами, досягне 91,88 мільярда доларів США до 2034 року, тоді як за іншою оцінкою, він становить 52,3 мільярда доларів США у 2025 році з прогнозом 130,2 мільярда доларів США до 2034 року.

Таке зростання відображає практичну реальність. Брандмауер, агент кінцевої точки, хмарна консоль або панель інструментів SIEM самі по собі не захищають бізнес. Хтось або щось повинно постійно збирати телеметрію безпеки, відрізняти реальну загрозу від очікуваної, досліджувати масштаби, стримувати загрозу та документувати те, що сталося. SOC – це операційна модель, яка робить ці дії повторюваними.

Центр операцій безпеки, або (SOC), – це централізована операційна функція для безперервного моніторингу безпеки, виявлення загроз, розслідування та реагування на інциденти. Вона може займати окреме приміщення, працювати через розподілену команду, через стороннього постачальника або поєднувати аналітиків з автоматизацією. Наявність приміщення необов'язкова. Операційна дисципліна – ні.

Центри моніторингу оборони (SOC) розвинулися з ранніх оборонно-орієнтованих команд моніторингу в середині 1970-х років і стали окремою функцією підприємства до кінця 1990-х та початку 2000-х років. Історія галузі пов'язує ранні концепції SOC з військовими та урядовими командними центрами, тоді як великі спалахи, такі як Code Red, Nimda та SQL Slammer, змусили великі підприємства, банки та операторів критичної інфраструктури централізувати моніторинг та реагування (історія та структура SOC). Важливим переходом стався зміщення від спеціального усунення несправностей в IT до спеціалізованої можливості постійного контролю.

Чотири функції, що визначають модель

– Моніторинг: Збір телеметрії з кінцевих точок, мереж, ідентифікаторів, програм, хмарних сервісів та засобів контролю безпеки.

– Виявлення: Застосовуйте правила кореляції, поведінковий аналіз та розвідку загроз для виявлення підозрілої активності.

– Розслідування: додавання контексту, реконструкція подій, визначення обсягу та відокремлення хибнопозитивних результатів від інцидентів.

– Відповідь: Локалізація, знищення, вилучення, передача та збереження доказів.

NIST описує SOC як центр захисту комп'ютерної мережі, що підтримується персоналом та інструментами, які безперервно об'єднують, співвідносять, аналізують та реагують на події, що стосуються безпеки, з різних джерел. Це визначення пояснює, чому SOC – це не просто черга сповіщень. Його цінність полягає в перетворенні гетерогенної телеметрії на пріоритетні, скоординовані дії.

SOC може мати велику ієрархію або невелику гібридну команду. Ні кількість персоналу, ні ліцензія на продукт не доводять, що організація має можливості SOC. Значущими тестами є безперервне покриття, задокументовані робочі процеси, вимірювана реакція та чітка відповідальність.

Моделі доставки SOC та як вибрати одну з них

Правильна модель SOC залежить не стільки від брендингу, скільки від того, хто відповідає за рішення у три критичні моменти: що розслідується, хто може це стримати та хто залишається відповідальним, коли реагування є незручним.

Внутрішній SOC надає організації максимальний контроль. Аналітики розуміють бізнес-системи, чутливість даних, поведінку користувачів та операційні пріоритети. Компроміс є суттєвим: організація повинна забезпечувати цілодобове покриття, фінансувати технологічний стек, підтримувати інженерію виявлення та зберігати дефіцитний досвід. Тиск на персонал є центральною проблемою. Уривок з опитування SOC за 2025 рік

показує, що 62% організацій кажуть, що вони не роблять достатньо для утримання найкращих талантів SOC, із середнім терміном служби 3-5 років.

Спільно керований SOC розподіляє обов'язки. Внутрішні співробітники можуть відповідати за розслідування високого ризику, комунікацію із зацікавленими сторонами та усунення наслідків, тоді як зовнішній постачальник займається постійним моніторингом, початковим сортуванням або пошуком загроз. Ця модель зберігає внутрішній контекст, але вимагає точної матриці відповідальності. Неоднозначні правила ескалації створюють затримки саме тоді, коли обидві команди припускають, що інша команда вжила заходів.

Повністю керована модель MDR або MSSP передає значну частину операцій з виявлення та реагування постачальнику. Вона пропонує швидше отримання вигоди та передбачуване операційне володіння, але клієнт відмовляється від деяких налаштувань і повинен перевіряти доступ, обробку доказів, повноваження на ескалацію та межі реагування.

Регуляторні обмеження також мають значення. Охорона здоров'я, фінанси, державні контракти та платіжні середовища можуть вимагати конкретних доказів, контролю доступу, практик зберігання та розподілу обов'язків. Перш ніж вибрати модель, задокументуйте системи щодо обсягу, необхідного покриття, повноважень реагування, зобов'язань щодо дотримання вимог та очікувань щодо передачі. Керовану послугу SOC можна оцінити відповідно до цих вимог, а не купувати як розпливчасту обіцянку моніторингу.

Люди, процеси та технології всередині SOC

Функціонуючий SOC має чіткі обов'язки, а не просто набір посад у сфері безпеки. Аналітики першого рівня переглядають вхідні сповіщення, додають базовий контекст, пригнічують відому доброякісну активність та передають підозрілі справжні позитивні результати. Аналітики другого рівня проводять глибше розслідування, співвідносять докази ідентифікації, кінцевих точок, мережі та хмари, а також визначають обсяг інциденту. Аналітики третього рівня та мисливці за загрозами проводять складні розслідування та проактивно шукають поведінку зловмисника, яку пропустили автоматичні виявлення.

Менеджер SOC відповідає за операційний ритм. Це включає охоплення змін, шляхи ескалації, якість набору завдань, пріоритети виявлення, звітність та координацію з IT-відділом, юридичним відділом, відділом конфіденційності та бізнес-зацікавленими сторонами. У менших організаціях одна людина може виконувати кілька ролей, але обов'язки все одно потребують чіткого визначення відповідальності.

Процес перетворює досвід на повторювані дії

Зрілий процес повідомляє аналітикам, що робити, перш ніж інцидент стане стресовим. Він охоплює сортування, визначення ступеня тяжкості, ескалацію, збереження доказів, затвердження стримування, перевірку відновлення та розгляд після інциденту. Передача змін повинна визначати відкриті розслідування, незавершені дії, уражені активи та власників бізнесу. Забезпечення якості повинно вибірково аналізувати закриті сповіщення та перевіряти, чи дійшли аналітики обґрунтованих висновків.

Технології посилюють цих людей та процеси:

– SIEM: Співвідносить події з кількох джерел і надає аналітикам спільну поверхню для розслідування.

– SOAR: Виконує методичні плани збагачення, видачі заявок, сповіщень та реагування.

– EDR: Забезпечує телеметрію процесів, файлів, користувачів та відповідей кінцевих точок.

– XDR: Розширює пов’язане виявлення та реагування на кінцеві точки, ідентифікаційні дані, електронну пошту, хмару та інші домени.

– NDR: Виявляє поведінку мережі, зокрема активність, яку елементи керування кінцевими точками не можуть спостерігати.

Стек дає збій, коли кожен інструмент створює окрему чергу. Оповіщення SIEM має надавати докази, необхідні для розслідування, SOAR має усувати повторювану обробку, а EDR або NDR повинні забезпечувати телеметричні та стримувальні дії, що усувають невизначеність. Люди залишаються

відповідальними за свої судження, особливо коли автоматизовані дії можуть порушити роботу критично важливої послуги.

Як працюють робочі процеси SOC від оповіщення до відновлення

О 02:14 правило кореляції SIEM позначає неможливий вхід у систему, пов’язаний з користувачем фінансового відділу. Сповіщення потрапляє до черги Tier-1 разом із доданою ідентифікатором, результатом автентифікації, географічним розташуванням джерела, інформацією про пристрій та нещодавньою активністю облікового запису. Аналітик перевіряє, чи подорожував користувач, чи розпізнано пристрій, і чи відбулися інші входи або зміни привілеїв приблизно під час тієї ж події.

Перше рішення – не «злом чи ні». Річ у тім, чи виправдовують наявні докази ескалацію. Рівень 1 визначає серйозність ситуації, записує обґрунтування та передає сповіщення Рівню 2, коли активність залишається підозрілою або впливає на конфіденційний обліковий запис.

Розслідування відокремлює підозру від об’єкта

Рівень 2 переходить від події ідентифікації до телеметрії EDR. Аналітик переглядає дерево процесів кінцевої точки, нещодавню активність автентифікації, контекст браузера та електронної пошти, а також пов'язані сповіщення. Докази свідчать про зловживання обліковими даними, але історія процесів кінцевої точки не підтверджує виконання або збереження. Ця відмінність змінює план стримування. Команда повинна захистити обліковий запис, не відключаючи критично важливу робочу станцію без потреби.

Рівень 3 та менеджер SOC координують дії через затверджений шлях реагування:

1. Скасувати активні сеанси користувача.

2. Примусова перереєстрація MFA.

3. Заблокуйте підозріле джерело на відповідному краю керування.

4. Ізолюйте кінцеву точку за допомогою EDR, якщо цього виправдовують докази.

5. Пошук пов’язаної активності у фінансових ідентифікаторах та системах.

6. Записуйте кожну дію, затвердження, позначку часу та результат у справі.

Потім інцидент проходить етап ліквідації та відновлення. Аналітики підтверджують, що несанкціоновані сеанси зникли, перевіряють кінцеву точку, відновлюють нормальний доступ у контрольованих умовах та відстежують повторення. Запис справи повинен містити початкове сповіщення, результати збагачення, запити, знімки екрана або посилання на докази, точки прийняття рішень, дії з стримування та схвалення відновлення. Практичний робочий процес реагування на інциденти безпеки може допомогти формалізувати ці передачі.

Після інциденту проводиться огляд, який визначає першопричину, прогалини у часових шкалах, слабкі місця у виявленні та схеми дій, що потребують перегляду. Команда може налаштувати правило неможливого переміщення, додати умову довіри до пристрою, покращити збагачення ідентифікаційних даних або змінити шлях ескалації для фінансових рахунків. Мета полягає не лише в тому, щоб закрити заявку. Мета полягає в тому, щоб зробити наступне розслідування швидшим та надійнішим.

Пояснення сучасного технологічного стеку SOC

Стек SOC слід розробляти як конвеєр виявлення, а не збирати як полицю непов'язаних продуктів. Перша проблема – це видимість. Якщо події ідентифікації, кінцевої точки, хмари, електронної пошти, DNS та мережі не збираються та не нормалізуються, подальша кореляція не зможе відновити відсутній контекст.

Посібник практикуючого спеціаліста групує джерела журналів за пріоритетами отримання. Рівень 1 включає журнали постачальників ідентифікаційних даних, телеметрію EDR, журнали хмарного аудиту, журнали шлюзу безпеки електронної пошти, запити DNS та дані брандмауера або мережевого потоку. Рівень 2 включає журнали WAF, проксі-сервера, VPN,

аудиту баз даних та SaaS. Рівень 3 включає інвентаризацію активів, результати вразливостей, розвідку загроз та дані HR для збагачення та кореляції.

Який внесок має вносити кожен шар

SIEM – це аналітичний центр. Він має співвідносити підозрілий вхід із станом пристрою, активністю кінцевих точок, змінами привілеїв та поведінкою мережі, а не генерувати окремі сповіщення для кожної події. UEBA може допомогти виявити активність, яка відхиляється від усталеної схеми користувача або організації, але аналітикам все одно потрібні докази, які можна пояснити.

SOAR має обробляти детерміновану роботу. Збагачення IP-адреси, отримання контексту ідентифікації, створення заяви, повідомлення власника або ізоляція хоста за затверджених умов є сильними кандидатами на автоматизацію. План дій, який вживає заходів без захисних огорож, може призвести до збою, тому повноваження реагування та кроки відкату мають бути включені до проектування.

EDR та XDR забезпечують глибину розслідування та охоплення реагування. NDR додає видимість там, де агенти кінцевих точок недоступні або де моделі трафіку виявляють горизонтальне переміщення. UTMStack об'єднує ці функції як платформу з відкритим кодом SIEM, SOAR та XDR, з обробкою журналів, кореляцією, UEBA, телеметрією кінцевих точок, мережевими датчиками та автоматизованими робочими процесами. Практичним випробуванням є те, чи може одне сповіщення пройти процес збору, кореляції, збагачення, розслідування та реагування, не змушуючи аналітиків копіювати докази між ізоляційними сховищами.

Ключові показники SOC та KPI, що мають значення

Менеджер SOC повинен переглянути показники, що описують швидкість, якість, охоплення, робоче навантаження та навчання. Кількість сповіщень сама по собі є слабким доказом. Команда може закрити багато сповіщень, пропустивши при цьому важливі дії, або зменшити обсяг, приховуючи виявлення, які повинні залишатися видимими.

Середній час виявлення (MTTD) вимірює, як швидко підозріла активність стає розпізнаним виявленням. Середній час реагування або відновлення (MTTR) вимірює час, необхідний для локалізації та відновлення уражених систем. Керівництво з автоматизації поєднує код, сценарії та робочі процеси на основі штучного інтелекту зі скороченням часу виявлення, сортування та локалізації (керівництво з автоматизації реагування на інциденти).

Довідник з операцій SOC з відкритим кодом містить чіткі цілі, які команди можуть використовувати як початкові механізми: MTTD менше 30 хвилин, MTTR менше 60 хвилин для критичних або високоризикових інцидентів, середній час підтвердження менше 10 хвилин, рівень хибнопозитивних результатів менше 10%, рівень закриття щонайменше 95% та завершення перевірки після інциденту протягом 72 годин для критичних або високоризикових подій.

Керівництво NIST щодо обробки інцидентів також підтримує відстеження загальної кількості витраченої праці на виявлення, аналіз, локалізацію, ліквідацію та відновлення, а не лише обсягу заявок. Додайте робоче навантаження аналітиків, вік невирішеної черги, охоплення телеметрією критичних активів та адаптацію джерел журналів, щоб виявити прогалини у видимості та вигоранні, перш ніж вони перетворяться на операційні збої.

Сучасні тенденції, що формують сучасну соціальну операційну систему

Хмарні та гібридні середовища змінили проблему збору даних. Аналітики тепер досліджують активність на різних площинах керування хмарою, SaaS-додатках, віддалених кінцевих точках, постачальниках ідентифікації та локальній інфраструктурі. SOC, який контролює лише центр обробки даних, може мати відшліфовану панель інструментів та суттєву сліпу зону.

Автоматизоване сортування стає практичною відповіддю на цю складність. Плейбуки можуть збагачувати сповіщення, перш ніж аналітики їх відкриють, отримувати пов'язані події, призначати власників та виконувати дії з низьким рівнем ризику. Робочі процеси з підтримкою LLM можуть

підсумовувати інцидент, складати пошукові запити, виявляти пов'язані сповіщення та пропонувати шляхи розслідування. Вони повинні підтримувати судження аналітиків, а не приймати незворотні рішення.

Консолідація та фокус на ідентичності

XDR відображає рух до підключеної телеметрії та скоординованого реагування. Консолідація може зменшити кількість розслідувань за принципом «обертальне крісло», але вона також може створити залежність від моделі даних та засобів керування реагуванням одного постачальника. Архітекторам слід протестувати варіанти експорту, прозорість виявлення, доступ до API та те, як платформа обробляє джерела поза межами своєї рідної екосистеми.

Ідентифікація стала центральною проблемою SOC, оскільки події автентифікації, привілеїв, сеансів та довіри пристроїв з’єднують користувачів з бізнес-доступом. Тому інженерія виявлення повинна розглядати телеметрію ідентифікації як першокласне джерело, поряд з даними кінцевих точок та мережі.

Відповідність також переходить у щоденну діяльність. Вимоги, пов'язані з PCI, HIPAA, SOC 2, ISO 27001, NIS2, CMMC та GLBA, легше підтримувати, коли докази, події доступу, записи відповідей та зіставлення елементів керування фіксуються під час звичайних робочих процесів, а не збираються під час аудиту. SOC не повинен створювати паралельний архів відповідності. Він повинен надавати обґрунтовані докази з тих самих розслідувань та заходів контролю, що використовуються для захисту навколишнього середовища.

Практичний висновок простий. Почніть з телеметрії та рішень щодо реагування, які є найважливішими, потім автоматизуйте стабільні робочі процеси та додайте розширений аналіз, коли базові дані є достовірними. Купівля шару штучного інтелекту перед виправленням інвентаризації активів, реєстрації ідентифікаційних даних або володіння зазвичай призводить до швидшого отримання зведень неповних доказів.

Працездатна дорожня карта SOC починається з прозорості, а не з командного центру. Почніть з визначення критичних активів, систем

ідентифікації, хмарних облікових записів, кінцевих точок, мережевих елементів керування та меж відповідності. Визначте, хто отримує ескалацію, а хто може санкціонувати стримування. Потім спочатку впровадьте найціннішу телеметрію, включаючи дані ідентифікації, EDR, хмарний аудит, електронну пошту, DNS та брандмауер.

Поетапний шлях впровадження

Етап 1, забезпечення видимості. Збирати та нормалізувати журнали з брандмауерів, кінцевих точок, хмарних сервісів та постачальників ідентифікаційних даних. Створити невеликий набір правил кореляції з високою достовірністю для компрометації облікових записів, зловживання привілеями, поведінки шкідливого програмного забезпечення та підозрілої мережевої активності. Переконатися, що аналітики можуть досліджувати сповіщення, не відкриваючи окрему консоль для кожної підтверджувальної події.

Етап 2, автоматизація повторюваного реагування. Додайте методичні посібники SOAR для боротьби з фішингом, атаками методом повного перебору та стримування програм-вимагачів. Почніть зі збагачення та сповіщення, а потім впровадьте контрольовані дії, такі як скасування сеансу, ізоляція кінцевих точок або обмеження облікового запису, після тестування процедур схвалення та відкату.

Етап 3, покращення контексту та пошуку. Додайте інформацію про загрози, UEBA, інформацію про вразливості, дані про активи та управління випадками. Використовуйте ці джерела для визначення пріоритетів виявлень щодо відкритих або критично важливих для бізнесу систем та для пошуку поведінки, яка не спричинила певне заздалегідь визначене правило.

Етап 4, впровадження вимірювань. Створення інформаційних панелей для MTTD, MTTR, підтвердження, хибнопозитивних результатів, закриття, завершення перевірки, покриття активів та навантаження аналітиків. Робочі процеси за допомогою LLM можуть узагальнювати випадки та допомагати налаштовувати правила, але кожна рекомендація повинна залишатися такою, що може бути переглянута.



Рисунок 3.1 – Структурна схема системи

Стислий тест на зрілість містить п'ять запитань:

– Видимість: Чи інтегровано критично важливі джерела ідентифікації, кінцевих точок, хмари, мережі та додатків?

– Виявлення: Чи охоплюють правила шляхи атак з найбільшим впливом на організацію?

– Автоматизація: Чи можуть аналітики послідовно збагачувати, маршрутизувати та стримувати поширені інциденти?

– Відповідь: Чи задокументовано повноваження, передачу, докази, стягнення та перегляд?

– Відповідність: Чи може команда надати відповідні контрольні докази з операційних записів?

Центральним рішенням залишається питання про те, чи працювати внутрішньо, розподіляти обов'язки з постачальником, чи використовувати MDR. Платформа повинна підтримувати цей вибір, а не диктувати його.

3.3 Розробка функціональної схеми

На рисунку 3.2 наведено функціональну схему розробленої системи. Згідно з архітектурою, комплекс складається з таких базових модулів:

– Блок виявлення дестабілізуючого трафіку – здійснює безперервний моніторинг вхідних потоків для фіксації аномалій і ознак кібератак.

– Блок визначення деструктивних дій та виду атаки – виконує класифікацію інцидентів (широкомовні шторми, ARP-spoofing тощо).

– Блок зберігання результатів – відповідає за безпечне накопичення логів, статистики та архівних даних.

– Блок моніторингу мережі – забезпечує загальний нагляд за станом мережевої інфраструктури.

– Блок аналізу мережної статистики – проводить глибокий розбір параметрів трафіку за допомогою екосистеми метрик.

– Блок визначення топології мережі – автоматично картографує зв'язки та ієрархію пристроїв.

Підсистема виявлення та усунення дестабілізуючого трафіку

Модуль виявлення та нейтралізації аномалій охоплює такі функціональні компоненти:

1. Блок визначення профілю нормального трафіку (навчання) – аналізує легітимну активність протягом базового періоду (наприклад, 24 години на добу впродовж тижня) для формування еталонних показників: інтенсивності пакетів (pps), співвідношення протокольних флагів (SYN/FIN) та кількості одночасних TCP-з'єднань на хост чи підмережу.

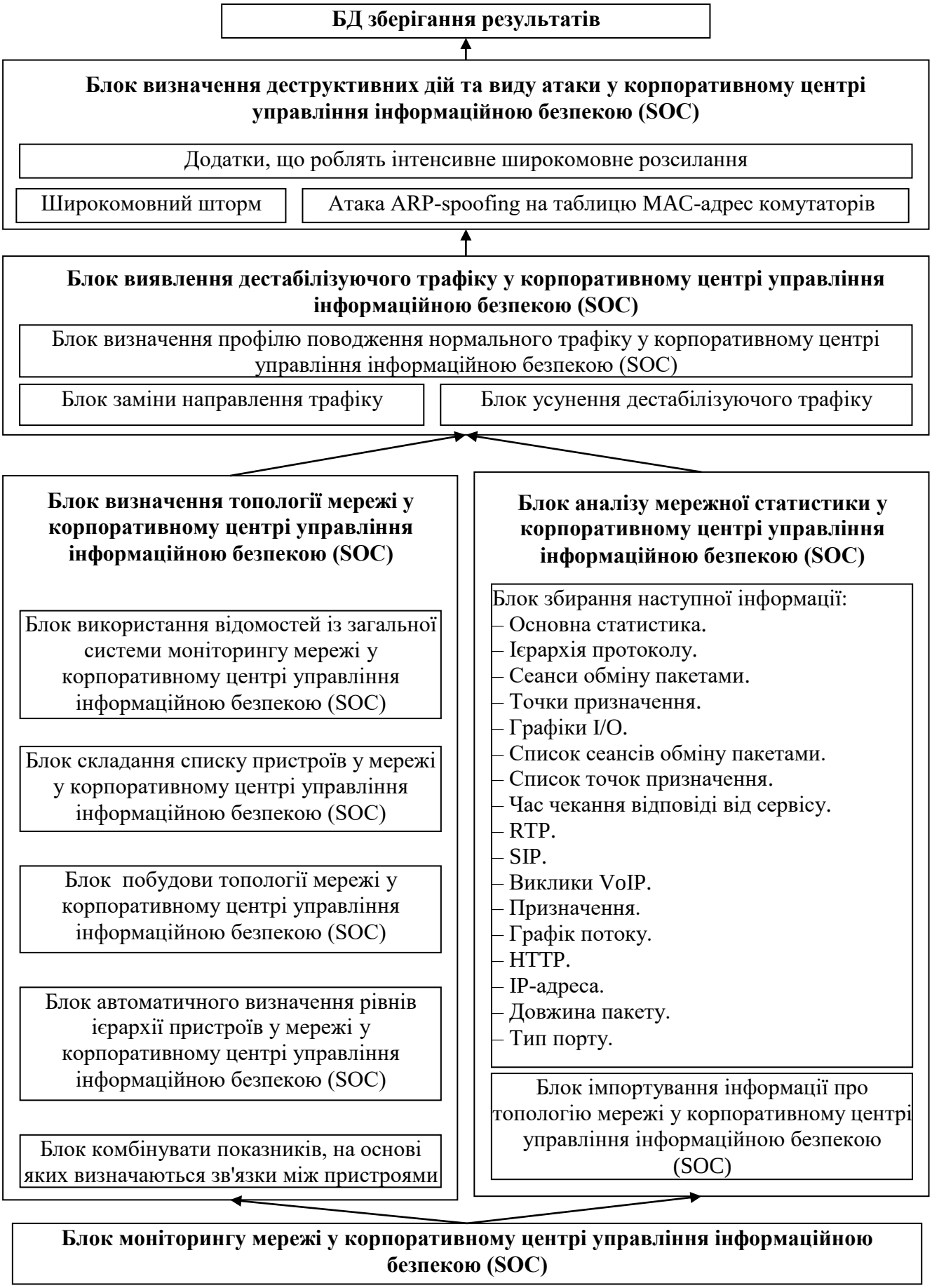

Рисунок 3.2 – Функціональна схема системи

2. Блок перенаправлення трафіку – реалізує механізми перехоплення потоків за допомогою протоколу BGP, Route Health Injection (RHI) або зовнішніх маршрутизаторів.

3. Блок усунення дестабілізуючого трафіку – виконує багатоступінчастий аналіз, застосовує контрзаходи проти спуфінгу, фільтрує шкідливі запити та здійснює ін'єкцію очищеного трафіку назад у штатний маршрут.

Класифікація деструктивних дій та атак

У межах модуля аналізу загроз виділяються такі типові дестабілізуючі фактори:

– Широкомовний шторм (Broadcast Storm): лавиноподібне розмноження широкомовних кадрів, що паралізує мережу. Відповідно до галузевих стандартів, частка широкомовного та багатоадресного трафіку не повинна перевищувати 8-10% від загального обсягу. Шторми часто виникають через петлі в мережі при збоях протоколу Spanning Tree (IEEE 802.1d) або навмисні дії зловмисників.

– ARP-spoofing: атака на канальному рівні (Layer 2) у мережах Ethernet, заснована на відправці підроблених (мимовільних) ARP-відповідей (gratuitous ARP). Це дозволяє зловмиснику видати себе за шлюз чи маршрутизатор, перехоплюючи та модифікуючи трафік між вузлами.

Блок аналізу мережної статистики

Для детального дослідження трафіку в межах SOC використовуються такі статистичні метрики та інструменти:

1. Основна статистика (Summary): властивості файлів захоплення, таймінги та параметри фільтрації.

2. Ієрархія протоколів: розбивка даних відповідно до моделі OSI.

3. Сеанси обміну (Conversations) та точки призначення (Endpoints): аналіз активних діалогів між хостами та обсягів переданих даних.

4. Графіки I/O (IO Graphs): візуалізація інтенсивності потоків за допомогою фільтрів відображення.

5. Аналіз мультимедійних протоколів (RTP, SIP, VoIP Calls): моніторинг

параметрів передачі голосу та відео в реальному часі.

6. HTTP-статистика: розподіл пакетів, лічильники та аналіз запитів.

7. Додаткові параметри: довжина пакетів, типи портів та розподіл IP-адрес.

Блок визначення топології мережі

Модуль картографування інфраструктури виконує такі завдання:

– Використання наявних відомостей із систем моніторингу без надмірного додаткового опитування обладнання.

– Автоматичне формування переліку мережевих пристроїв.

– Побудова хронологічної топології зі створенням зрізів на задану дату та відстеженням змін.

– Автоматична ієрархічна класифікація вузлів на периферійні, проміжні та центральні.

– Статистична перевірка значущості зв'язків між елементами незалежно від апаратних платформ.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

На рисунку 3.3 наведено діаграму процесів розробленої системи, детальний розгляд якої дозволяє простежити логіку взаємодії її функціональних компонентів. У процесі проєктування використовується структурний підхід та графічне представлення інформаційних потоків (Data Flow Diagram, DFD), що дає змогу ефективно візуалізувати процеси обробки та передачі даних у межах комплексу.

Згідно з інженерною методологією, розробка розпочинається з побудови контекстної діаграми (діаграми нульового рівня), яка визначає глобальні межі системи та її інтерфейси із зовнішнім середовищем. Надалі ця модель підлягає ітераційній декомпозиції шляхом деталізації базових процесів та потоків даних з

метою повного відображення архітектури розроблюваного програмно-апаратного рішення.

Складені діаграми потоків даних базуються на чотирьох основних типах елементів:

– Процеси – функціональні перетворення даних у межах описуваної системи (наприклад, збір телеметрії, аналіз пакетів, класифікація загроз чи генерація управляючих команд).

– Сховища даних (репозиторії) – структуровані сховища для безпечного збереження логів, баз сигнатур та архівних даних.

– Зовнішні сутності – джерела та споживачі інформації поза межами системи (користувачі, адміністратори, зовнішні сервіси чи мережеві вузли).

– Потоки даних – канали передачі інформаційних масивів між попередніми трьома типами елементів.



Рисунок 3.3 – Діаграма взаємодії процесів

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Функціонування системи мережевого центру управління інформаційною безпекою SOC з модулями автоматизованого реагування та оркестрації інцидентів ґрунтується на централізованому збиранні подій безпеки, їхньому аналізі, виявленні потенційних загроз та виконанні відповідних захисних дій. Алгоритмічне забезпечення системи повинно організовувати безперервний процес оброблення телеметричних даних від мережевого обладнання, серверів, робочих станцій і засобів інформаційної безпеки.

Особливе значення має узгодження механізмів виявлення інцидентів із процедурами автоматизованого реагування. Це дозволяє формувати керуючі рішення на основі отриманих подій та встановлених політик безпеки. Загальний алгоритм функціонування програмного комплексу представлено на рисунку 4.1.

Основою проєктного рішення є поєднання функціональності SIEM та SOAR у межах єдиної системи керування інформаційною безпекою.

Підсистема SIEM забезпечує приймання, нормалізацію, збереження та кореляцію подій. Підсистема SOAR відповідає за координацію захисних операцій, виконання сценаріїв реагування та взаємодію із зовнішніми програмними засобами.

Модуль керування інцидентами об'єднує результати аналізу, відомості про захищені ресурси та історію виконаних дій. Запропонована архітектура передбачає логічне відокремлення аналітичних функцій від компонентів, які мають повноваження змінювати конфігурацію мережевих пристроїв та інформаційних систем.

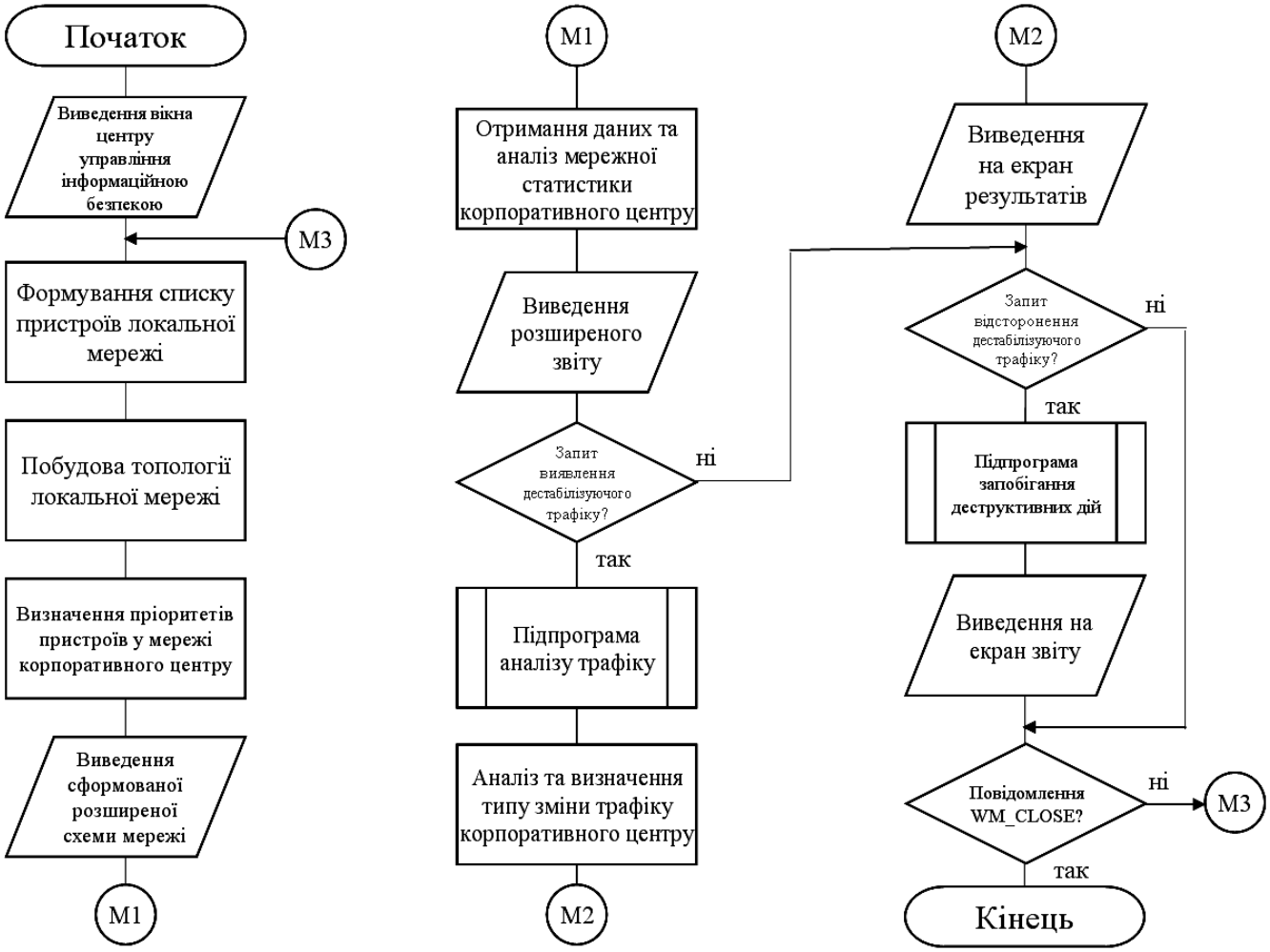

Рисунок 4.1 – Блок-схема основної програми

Оброблення подій починається з отримання повідомлень від контрольованих джерел. Надалі виконується перевірка їхньої структури, визначення часових характеристик, ідентифікація вузлів та приведення даних до внутрішнього формату. Після цього події передаються до модуля кореляції, який встановлює наявність взаємопов'язаних ознак небезпечної активності. Якщо задані умови виконуються, формується повідомлення про виявлення загрози. Воно використовується для створення або оновлення інциденту, визначення пріоритету та вибору допустимого сценарію реагування.

Для автоматизації прийняття захисних рішень доцільно передбачити окремий модуль оркестрації інцидентів. Його завдання полягає у визначенні послідовності необхідних операцій, перевірці умов запуску та контролі

результатів виконання. Вхідними даними є характеристики інциденту, результати кореляції, контекст захищеного ресурсу та встановлені обмеження автоматизації. Вихідним результатом стає сформований план реагування або виконана послідовність дозволених захисних дій. Роботу відповідної підпрограми оркестрації відображено на рисунку 4.2.

Важливо розмежовувати поняття події безпеки, сповіщення та інциденту. Подія характеризує окрему зміну стану інформаційної системи. Сповіщення формується після спрацювання правила виявлення або аналітичного механізму. Інцидент являє собою встановлену або обґрунтовано підозрювану сукупність подій, яка потребує реагування. Один інцидент може містити декілька сповіщень із різних джерел. Тому алгоритмічне забезпечення повинно передбачати об'єднання пов'язаних повідомлень та запобігання створенню надмірної кількості дубльованих інцидентів.

Розроблення такого програмного комплексу потребує узгодження алгоритмів оброблення телеметрії, виявлення підозрілої активності, пріоритизації інцидентів, керування сценаріями реагування та контролю виконання операцій. Як можливу основу програмної реалізації можна розглядати Python для серверної логіки, структуровані повідомлення JSON для обміну даними, реляційну базу даних для обліку інцидентів і програмні інтерфейси REST API для взаємодії із засобами захисту. Конкретний склад технологій має визначатися фактичною реалізацією. Подальший опис характеризує технічно здійсненну проєктну модель, а не непідтверджені результати впровадження.

Організація потокового приймання та уніфікації подій безпеки

Першим функціональним етапом роботи SOC є отримання телеметричних повідомлень від контрольованих інформаційних ресурсів.

Джерелами можуть бути міжмережеві екрани, маршрутизатори, сервери автентифікації, операційні системи, засоби захисту кінцевих пристроїв та мережеві системи виявлення вторгнень. Кожен тип джерела формує повідомлення відповідно до власної структури. Для подальшого автоматизованого аналізу необхідно привести ці повідомлення до спільного набору полів.

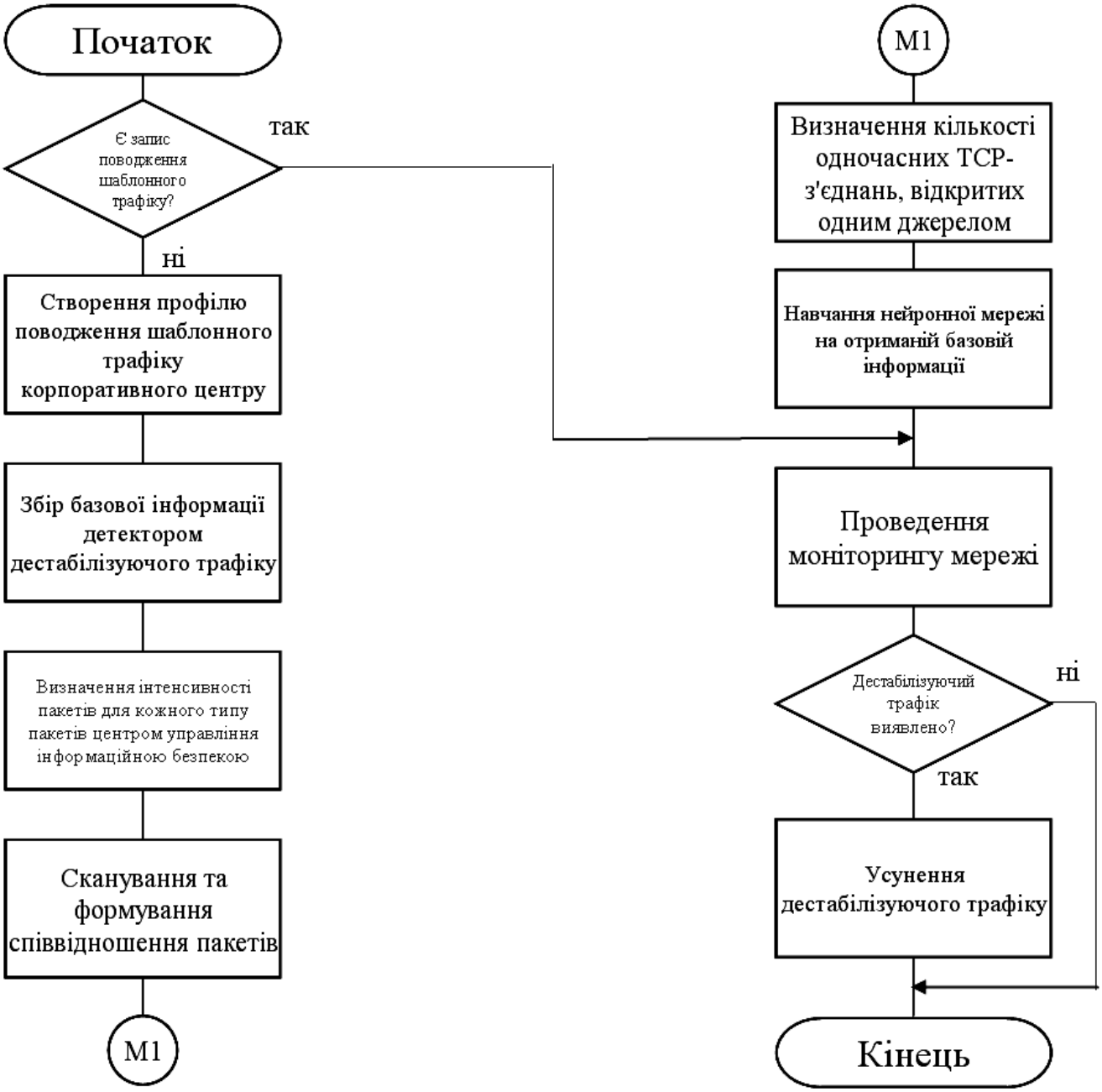

Рисунок 4.2 – Блок-схема роботи підпрограми

Збирання подій доцільно організувати через окремий модуль приймання. Він повинен підтримувати отримання повідомлень із системних журналів, службових агентів або програмних інтерфейсів. Для мережевих пристроїв можливе використання Syslog.

Для програмних сервісів доцільно передбачити приймання структурованих повідомлень у форматі JSON. Передавання інформації від зовнішніх агентів

повинно здійснюватися через контрольовані мережеві інтерфейси. За наявності відповідної підтримки необхідно використовувати захищені канали та автентифікацію джерел.

Після отримання повідомлення виконується перевірка його відповідності очікуваному формату. Алгоритм визначає наявність обов'язкових полів, допустимість типів даних та коректність часової позначки. Повідомлення з помилковою структурою не повинно передаватися до основного механізму кореляції без попереднього оброблення. Для таких випадків доцільно передбачити окрему чергу помилкових записів із зазначенням причини відхилення. Це дозволяє контролювати проблеми інтеграції та запобігати втраті діагностичної інформації.

Нормалізація полягає у формуванні внутрішнього представлення події, незалежного від формату джерела. Доцільно передбачити поля event_id, timestamp, source, event_type, severity, host_id, src_ip, dst_ip та attributes. Ідентифікатор event_id використовується для контролю повторної доставки. Поле source визначає походження повідомлення, а event_type характеризує його семантичну категорію. Додаткові атрибути дозволяють зберігати специфічні характеристики окремих засобів захисту без зміни основної структури.

Часові позначки повинні приводитися до узгодженої часової шкали, наприклад UTC. Це необхідно для коректного зіставлення повідомлень від пристроїв, розташованих у різних часових зонах. При цьому алгоритм повинен розрізняти час виникнення події та час її отримання сервером SOC. Різниця між ними використовується для виявлення затримок доставки. Якщо джерело має некоректно налаштований системний час, кореляція подій може призвести до помилкових результатів.

Для програмного представлення нормалізованої події можна використати структуру dataclass у Python. Вона дозволяє описати набір полів та контролювати їхнє використання іншими модулями. Наведений фрагмент демонструє варіант перевірки вхідного повідомлення та формування уніфікованої структури.

```
from dataclasses import dataclass
from datetime import datetime, timezone
```

```
from ipaddress import ip_address
from typing import Any

# Уніфіковане представлення події безпеки SOC
@dataclass(frozen=True)
class SecurityEvent:
    event_id: str
    timestamp: datetime
    source: str
    event_type: str
    severity: int
    host_id: str
    src_ip: str | None
    attributes: dict[str, Any]
def normalize_event(payload: dict) -> SecurityEvent:
    # Перевірка обов'язкових текстових полів
    required = ("event_id", "source", "event_type", "host_id")

    for field in required:
        value = payload.get(field)
        if not isinstance(value, str) or not value.strip():
            raise ValueError(f"Некоректне поле: {field}")
    # Перетворення часової позначки на UTC
    raw_time = payload.get("timestamp")
    if not isinstance(raw_time, str):
        raise ValueError("Відсутня часова позначка")

    event_time = datetime.fromisoformat(
        raw_time.replace("Z", "+00:00")
    )
    if event_time.tzinfo is None:
        raise ValueError("Часова зона не визначена")
   event_time = event_time.astimezone(timezone.utc)
    # Перевірка допустимого рівня критичності
    severity = payload.get("severity")
    if type(severity) is not int or not 0 <= severity <= 10:
        raise ValueError("Некоректний рівень критичності")

    # Перевірка адреси джерела, якщо вона задана
    src_ip = payload.get("src_ip")
    if src_ip is not None:
        src_ip = str(ip_address(src_ip))

    attributes = payload.get("attributes", {})
    if not isinstance(attributes, dict):
        raise ValueError("Некоректні атрибути події")
    return SecurityEvent(
        event_id=payload["event_id"],
        timestamp=event_time,
        source=payload["source"],
        event_type=payload["event_type"],
        severity=severity,
        host_id=payload["host_id"],
        src_ip=src_ip,
        attributes=attributes.copy()
    )
```

Наведений код демонструє механізм приведення вхідних даних до внутрішнього формату. Якщо повідомлення містить некоректну часову позначку, адресу або рівень критичності, генерується виняток.

У складі серверного застосунку цей виняток повинен перехоплюватися модулем приймання. Після цього відповідна подія реєструється як помилкова без припинення оброблення інших повідомлень. Для промислового застосування додатково необхідно передбачити обмеження розміру повідомлення, перевірку схеми та контроль автентичності джерела.

Наступним етапом є усунення повторних повідомлень. Дублювання може виникати через повторне передавання агентом, відновлення мережевого з'єднання або повторне читання журналу після перезапуску сервісу. Для контролю використовується комбінація ідентифікатора джерела та event_id. Якщо такий запис уже існує, повторне повідомлення не повинно створювати нову подію. Для джерел без власних ідентифікаторів необхідно визначати окремий механізм дедуплікації, який враховує можливість існування різних подій з однаковим текстовим вмістом.

Завершення нормалізації передбачає передавання події до сховища та модуля кореляції. Для розподіленої архітектури між ними може використовуватися брокер повідомлень. Він забезпечує буферизацію та незалежне оброблення потоків даних. При цьому факт доставки повідомлення до брокера не повинен ототожнюватися з успішним завершенням кореляції. Підтвердження оброблення необхідно формувати після збереження результату або виконання визначеної транзакційної операції.

Кореляція телеметрії та формування інцидентів SOC

Основним завданням кореляційного механізму є встановлення зв'язків між подіями, які окремо можуть не свідчити про порушення безпеки. Наприклад, одинична невдала спроба автентифікації не завжди є ознакою атаки. Проте велика кількість таких спроб протягом короткого інтервалу може характеризувати автоматизований підбір облікових даних. Якщо після них зафіксовано успішну автентифікацію з того самого джерела, виникають додаткові підстави для перевірки активності.

Для виявлення подібних ситуацій доцільно використовувати кореляційні правила, які враховують тип події, часовий інтервал, джерело та кількість

повторень. Вхідними даними є нормалізовані повідомлення, впорядковані відповідно до прийнятої політики оброблення часу. Результатом стає сповіщення з ідентифікатором правила, переліком пов'язаних подій і характеристиками виявленої активності.

```
from collections import defaultdict, deque
from datetime import timedelta


class AuthenticationCorrelator:
    def __init__(self, window_seconds=300, threshold=5):
        # Тривалість вікна та поріг спрацювання
        self.window = timedelta(seconds=window_seconds)
        self.threshold = threshold

        # Часові позначки групуються за джерелом
        # та контрольованим обліковим записом
        self.attempts = defaultdict(deque)
        self.last_seen = {}
        self.triggered = set()

    def process(self, event: SecurityEvent) -> bool:
        if event.event_type != "authentication_failed":
            return False
        account = event.attributes.get("account")
        if not isinstance(account, str) or not account:
            return False
        if not event.src_ip:
            return False
        key = (event.source, event.src_ip, account)
        now = event.timestamp
        # Перевірка порядку надходження подій
        previous = self.last_seen.get(key)
        if previous is not None and now < previous:
            raise ValueError("Порушено часовий порядок")
        self.last_seen[key] = now
        queue = self.attempts[key]
        # Видалення подій поза часовим вікном
        while queue and now - queue[0] > self.window:
            queue.popleft()
        queue.append(now)

        # Одне сповіщення до виходу нижче порогу
        if len(queue) >= self.threshold:
            if key not in self.triggered:
                self.triggered.add(key)
                return True
        else:
            self.triggered.discard(key)

        return False
```

Цей алгоритм визначає наявність встановленої кількості невдалих спроб автентифікації в межах заданого інтервалу. Результат True означає необхідність формування сповіщення, а не остаточне підтвердження компрометації облікового запису. Порогові значення повинні визначатися під час налаштування системи.

Для практичного застосування необхідно також передбачити очищення неактивних кореляційних ключів, відновлення стану після перезапуску та врахування повторно доставлених подій.

Складніші правила можуть поєднувати декілька типів повідомлень. Наприклад, послідовність невдалих автентифікацій, успішний вхід і подальше виконання адміністративних операцій може утворювати окремий сценарій кореляції. У такому випадку необхідно контролювати не тільки кількість подій, а й порядок їх виникнення. Для кожного етапу сценарію доцільно визначати допустимий часовий інтервал та перелік умов переходу.

Кореляційний механізм повинен також враховувати контекст захищених ресурсів. Ідентична подія може мати різну значущість залежно від того, чи стосується вона тестового сервера, робочої станції або критичного вузла автентифікації. Для цього до повідомлення додаються характеристики активу, отримані з реєстру інфраструктури. Відомості про критичність, власника ресурсу та належність до мережевого сегмента використовуються під час наступного етапу оцінювання інциденту.

Об'єднання пов'язаних сповіщень повинно виконуватися за однозначно визначеними правилами. Доцільно враховувати ідентифікатор вузла, категорію загрози, часовий інтервал і спільні індикатори компрометації. Якщо відповідний відкритий інцидент уже існує, нове сповіщення додається до нього. Якщо зв'язок не встановлено, створюється окремий запис. При цьому система повинна зберігати посилання на вихідні події, щоб аналітик міг перевірити підстави для кореляції.

Проєктне рішення щодо оркестрації автоматизованих захисних операцій

Модуль SOAR призначений для координації процедур реагування між різними засобами інформаційної безпеки. Його функціонування базується на використанні сценаріїв реагування, які визначають послідовність дій, умови виконання та правила оброблення помилок. Такі сценарії часто називають

playbook. Вони дозволяють формалізувати типові операції аналітика SOC та виконувати їх у визначеному порядку.

Під час розроблення структури сценаріїв можна враховувати підходи специфікації CACAO Security Playbooks Version 2.0. Вона передбачає формалізований опис кроків, умовних переходів та керуючих операцій у сценаріях реагування. Використання подібного підходу дозволяє відокремити логіку процесу від конкретних команд зовнішніх систем. При цьому повна сумісність із CACAO потребує реалізації встановленої специфікацією структури даних і не випливає лише з наявності механізму сценаріїв.

Початковим етапом оркестрації є вибір сценарію відповідно до категорії інциденту. Наприклад, для підозрілої мережевої активності може застосовуватися сценарій перевірки IP-адреси та обмеження мережевих з'єднань. Для можливої компрометації облікового запису доцільно передбачити перевірку історії автентифікації, активних сеансів і адміністративних привілеїв. Для виявлення шкідливого програмного забезпечення сценарій може містити ізоляцію кінцевого пристрою та запуск додаткового сканування. Набір допустимих операцій залежить від функціональності підключених засобів захисту.

Після вибору сценарію виконується перевірка передумов. Алгоритм визначає доступність потрібного програмного інтерфейсу, наявність відповідних повноважень та коректність цільових параметрів. Додатково перевіряється належність ресурсу до переліку критичних об'єктів, для яких автоматичні зміни заборонені або обмежені. Якщо необхідна умова не виконується, сценарій не повинен переходити до виконання захисних команд. Замість цього формується повідомлення про неможливість автоматичного реагування.

Процедури реагування доцільно розділити на автоматичні, напівавтоматичні та ручні. Автоматичні операції виконуються без участі аналітика за наявності визначених умов. Напівавтоматичні потребують підтвердження відповідальної особи. Ручні процедури виконуються оператором SOC із використанням отриманих системою рекомендацій. Вибір режиму

залежить не лише від критичності загрози, а й від потенційних наслідків захисної операції.

Для прикладу автоматичне створення картки інциденту та збирання додаткової телеметрії зазвичай мають менший операційний ризик, ніж відключення сервера або блокування корпоративного облікового запису. Тому перші операції можуть виконуватися без окремого підтвердження за відповідної політики. Для дій, здатних порушити доступність критичних сервісів, необхідно передбачити адміністративне погодження. Такий підхід дозволяє обмежити наслідки помилкових спрацювань кореляційного механізму.

Програмну логіку вибору режиму реагування можна реалізувати окремою функцією. Вона отримує категорію інциденту, оцінку достовірності, характеристику ресурсу та параметри політики. Результатом стає об'єкт із допустимими діями та ознакою необхідності погодження. Наведений приклад демонструє формування рішення для блокування мережевого джерела. Він не виконує змін у міжмережевому екрані та призначений для подальшої інтеграції з авторизованим виконавчим модулем.

```
from dataclasses import dataclass
from ipaddress import ip_address

# Характеристики інциденту для модуля SOAR
@dataclass(frozen=True)
class IncidentContext:
    incident_id: str
    source_ip: str
    confidence: float
    severity: int
    critical_asset: bool
    confirmed: bool

# Результат попереднього планування реагування
@dataclass(frozen=True)
class ResponsePlan:
    incident_id: str
    action: str
    target: str
    requires_approval: bool
    reason: str

def plan_response(
    incident: IncidentContext,
    protected_addresses: set[str]
) -> ResponsePlan:
    # Перевірка коректності характеристик
    if not 0 <= incident.confidence <= 1:
        raise ValueError("Некоректна достовірність")
    if not 0 <= incident.severity <= 10:
```

```
        raise ValueError("Некоректна критичність")
    target_ip = ip_address(incident.source_ip)
    # Критичні адреси не блокуються автоматично
    protected = {
        str(ip_address(address))
        for address in protected_addresses
    }
    if incident.critical_asset or str(target_ip) in protected:
        return ResponsePlan(
            incident.incident_id,
            "manual_review",
            str(target_ip),
            True,
            "Захищений або критичний ресурс"
        )
    # Непідтверджені інциденти потребують аналізу
    if not incident.confirmed:
        return ResponsePlan(
            incident.incident_id,
            "investigate",
            str(target_ip),
            False,
            "Інцидент потребує підтвердження"
        )
    # Автоматичне блокування лише за заданих умов
    if incident.confidence >= 0.9 and incident.severity >= 8:
        return ResponsePlan(
            incident.incident_id,
            "block_ip",
            str(target_ip),
            False,
            "Висока достовірність і критичність"
        )
    return ResponsePlan(
        incident.incident_id,
        "request_approval",
        str(target_ip),
        True,
        "Необхідне рішення аналітика SOC"
    )
```

У наведеному прикладі використано умовні пороги достовірності та критичності для демонстрації принципу роботи. Їхні фактичні значення повинні визначатися політикою безпеки. Навіть за виконання числових умов виконавчий модуль повинен додатково перевірити авторизацію, допустимість адреси, область дії блокування та актуальний стан інциденту. Ознака requires_approval використовується для заборони передавання команди до зовнішнього засобу захисту без відповідного погодження.

Виконання сценарію може передбачати взаємодію з декількома програмними засобами. Наприклад, після виявлення підозрілої активності система отримує додаткові відомості від сервера журналювання, створює запис інциденту та надсилає команду мережевому засобу захисту. Кожна операція повинна мати

власний ідентифікатор, часові обмеження та очікуваний результат. Це дозволяє визначити, на якому етапі сценарій завершився успішно або виникла помилка.

Для доступу до зовнішніх сервісів доцільно використовувати окремі програмні адаптери. Кожен адаптер відповідає за автентифікацію, формування запиту, перевірку відповіді та перетворення результату до внутрішнього формату. Такий підхід дозволяє змінювати конкретний засіб захисту без істотного перепроєктування сценаріїв реагування. При цьому адаптери повинні мати мінімально необхідні права, а облікові дані мають зберігатися окремо від програмного коду.

Особливе значення має повторне виконання операцій. Якщо зовнішній сервіс виконав команду, але відповідь не надійшла через мережеву помилку, повторне надсилання може створити дубльовану дію. Для запобігання цьому необхідно використовувати ідентифікатори операцій та механізми перевірки фактичного стану зовнішньої системи. Якщо програмний інтерфейс підтримує ключі ідемпотентності, їх доцільно застосовувати для повторних запитів. За відсутності такої підтримки необхідно виконувати контроль стану перед повторною операцією.

Для кожного сценарію потрібно визначити поведінку за часткового виконання. Наприклад, якщо блокування мережевого джерела виконано успішно, а створення службового повідомлення завершилося помилкою, автоматичне скасування блокування не завжди є правильним рішенням. Алгоритм повинен враховувати значущість виконаної операції та встановлені правила компенсації. Для оборотних змін може передбачатися відновлення початкової конфігурації. Для незворотних дій необхідно використовувати окремі процедури контролю та погодження.

Керування життєвим циклом інциденту та виконанням сценаріїв

Для узгодження роботи аналітичних і виконавчих модулів доцільно використовувати формалізовану модель станів інциденту. Початковим станом є NEW, який встановлюється після створення запису. Після первинної перевірки інцидент переходить до TRIAGED.

Подальший аналіз виконується у стані INVESTIGATING. Після прийняття рішення про захисні операції використовується RESPONDING. Завершення реагування переводить інцидент до MONITORING, а після підтвердження відсутності необхідності в подальших діях він може отримати стан CLOSED.

Окремо необхідно передбачити можливість визначення помилкового спрацювання. Для цього може використовуватися стан FALSE_POSITIVE, який встановлюється після перевірки відповідальним аналітиком або за формально дозволеною процедурою. Такий результат повинен зберігатися в історії інциденту. Він може використовуватися під час уточнення правил кореляції, але не повинен автоматично змінювати інші активні інциденти з подібними ознаками.

Переходи між станами повинні виконуватися відповідно до визначених умов. Наприклад, перехід до RESPONDING допускається після вибору сценарію та проходження необхідних перевірок. Якщо сценарій потребує погодження, інцидент може перебувати в окремому стані очікування. Перехід до CLOSED не повинен виконуватися лише на підставі успішного надсилання захисної команди. Необхідно підтвердити виконання операції та завершення передбачених процедур перевірки.

Для сценаріїв реагування доцільно використовувати власні стани PENDING, RUNNING, WAITING_APPROVAL, SUCCEEDED, FAILED та CANCELLED. Це дозволяє незалежно контролювати життєвий цикл інциденту та виконання конкретного сценарію. Один інцидент може містити кілька послідовних сценаріїв, наприклад збирання додаткових доказів, локалізацію загрози та перевірку відновлення працездатності.

Керування станами потребує захисту від одночасного виконання суперечливих операцій. Якщо два робочі процеси намагаються змінити один інцидент, система повинна забезпечити узгодженість результату. Для цього можуть використовуватися транзакції бази даних та оптимістичне блокування за номером версії запису. Якщо версія змінилася після читання, операція повинна повторно перевірити актуальні умови переходу.

У базі даних доцільно зберігати окремі сутності інцидентів, сповіщень, вихідних подій, сценаріїв реагування та результатів виконання операцій. Між ними встановлюються зв'язки за унікальними ідентифікаторами. Така організація дозволяє відновити повну послідовність оброблення інциденту та встановити, які саме події спричинили запуск відповідного сценарію.

Для збереження доказової інформації необхідно передбачити журнал аудиту. Він повинен містити відомості про зміну станів, застосовані політики, виконавців операцій та результати звернень до зовнішніх систем. Часові позначки записів приводяться до узгодженого формату. Зміна або видалення журналів повинні обмежуватися відповід1но до встановленої моделі доступу. Для особливо значущих записів може використовуватися додатковий контроль цілісності.

Процедури реагування доцільно узгоджувати із загальною моделлю керування інцидентами. Зокрема, NIST SP 800-61 Revision 3 розглядає реагування на інциденти як частину ширшого процесу управління кібербезпековими ризиками. У межах запропонованого програмного комплексу це означає необхідність пов'язувати виявлення, аналіз, захисні операції та відновлення з відповідними управлінськими процедурами.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення мережевого центру управління інформаційною безпекою SOC призначене для централізованого збирання та аналізу подій безпеки, виявлення інцидентів і координації автоматизованих захисних операцій. Система об'єднує функції моніторингу мережевої інфраструктури, кореляції телеметричних повідомлень та оркестрації сценаріїв реагування SOAR. Її використання передбачає взаємодію із серверним обладнанням, мережевими пристроями та зовнішніми засобами інформаційної безпеки. Головне вікно програми показано на рисунку 5.1.

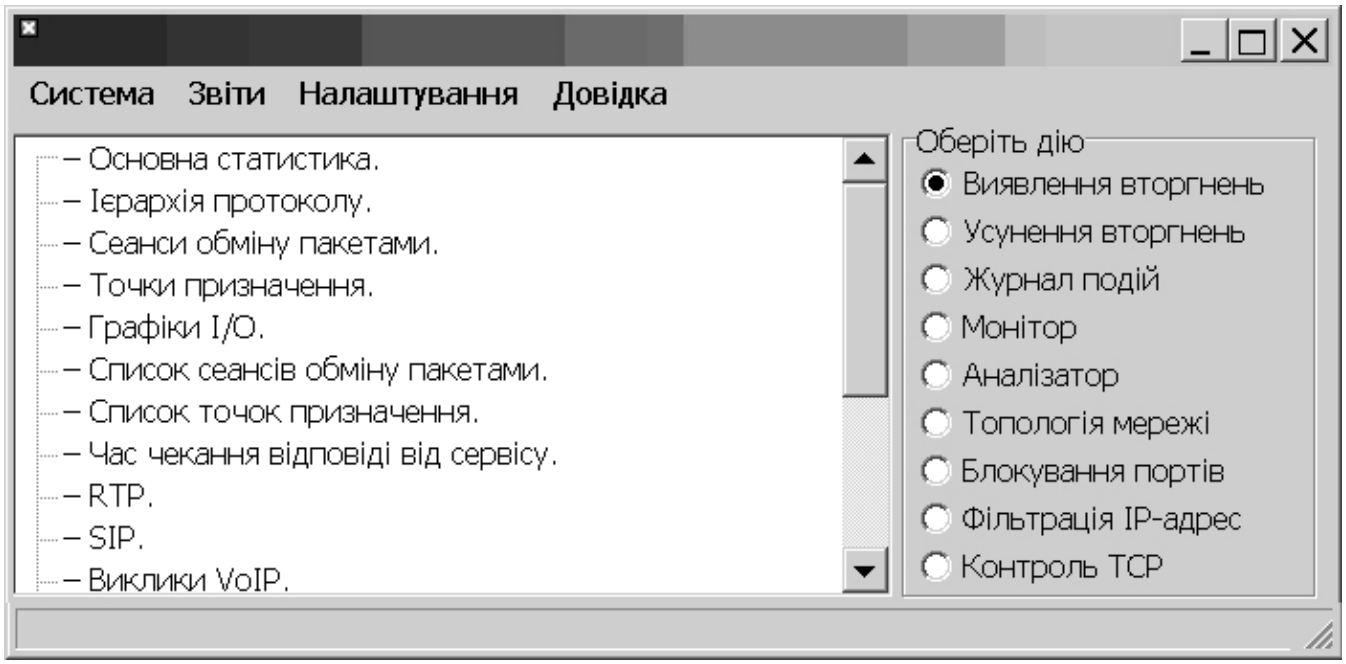

Рисунок 5.1 – Головне вікно ПЗ

Для впровадження запропонованої системи необхідно підготувати серверне середовище, яке забезпечує приймання телеметрії, зберігання подій, виконання кореляційних правил та взаємодію із засобами захисту. Як основу

розгортання доцільно використовувати серверну операційну систему Linux. Вона забезпечує необхідні засоби мережевої взаємодії, керування службами та ізоляції програмних процесів. Компоненти SOC можуть розміщуватися на одному сервері або розподілятися між кількома віртуальними машинами. Вибір конфігурації залежить від інтенсивності надходження подій і кількості контрольованих ресурсів.

Для серверної логіки запропонованої архітектури передбачається використання Python. Засобами цієї мови можуть реалізовуватися модулі нормалізації телеметрії, кореляції подій, оцінювання критичності інцидентів та керування сценаріями SOAR. Обмін структурованими повідомленнями доцільно організувати у форматі JSON. Для взаємодії із зовнішніми інформаційними системами можуть використовуватися REST API. Збереження інцидентів, станів сценаріїв і результатів виконання операцій доцільно реалізувати в реляційній базі даних, наприклад PostgreSQL. Зазначені технології становлять рекомендовану конфігурацію, а їхній остаточний склад визначається програмною реалізацією.

Під час підготовки обчислювального середовища необхідно перевірити доступність процесорних ресурсів, оперативної пам'яті та дискової підсистеми. Найбільший обсяг сховища зазвичай потрібен для накопичення телеметричних повідомлень і журналів аудиту. Обсяг необхідних ресурсів залежить від кількості джерел, середньої інтенсивності надходження подій та встановленого строку їх зберігання. Для розподіленого розгортання додатково необхідно забезпечити стабільне мережеве з'єднання між компонентами SOC. Системний час серверів і контрольованих пристроїв повинен бути синхронізований.

Підготовка програмного середовища розпочинається зі встановлення сумісного інтерпретатора Python та необхідних бібліотек. Для ізоляції залежностей доцільно створити окреме віртуальне середовище виконання. Після його активації встановлюються бібліотеки для мережевої взаємодії, перевірки структурованих даних, підключення до бази даних та оброблення асинхронних завдань. Конкретні версії компонентів повинні відповідати залежностям

програмного проєкту. Після встановлення необхідно перевірити коректність імпорту програмних модулів і доступність службових інтерфейсів.

Окремою складовою програмного продукту є відомості про його авторство. Вони призначені для ідентифікації розробника та зв'язку представленого програмного рішення з магістерською кваліфікаційною роботою. Відображення таких даних доцільно передбачити в інформаційному розділі програми без зазначення конфіденційних параметрів експлуатаційного середовища. Авторські дані розробленого програмного забезпечення представлено на рисунку 5.2.

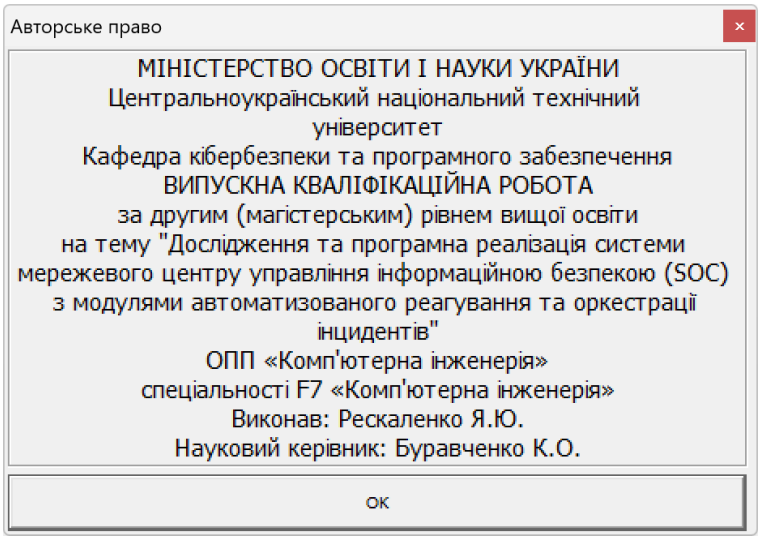

Рисунок 5.2 – Шаблона форма авторського права

Наступним етапом є підготовка сховища інформації. У базі даних необхідно передбачити структури для збереження нормалізованих подій, сповіщень, інцидентів, сценаріїв реагування та журналів виконаних операцій. Для кожної сутності визначаються унікальні ідентифікатори та зв'язки з іншими записами. Окремо налаштовуються облікові записи програмних компонентів і права доступу до відповідних таблиць. Перед початковим запуском необхідно перевірити можливість встановлення з'єднання, створення записів та виконання операцій читання.

Для організації приймання телеметрії потрібно налаштувати джерела повідомлень. Мережеві пристрої можуть передавати журнали за протоколом Syslog, а програмні агенти використовувати структуровані повідомлення через захищені програмні інтерфейси. Для кожного джерела визначаються мережеві адреси, дозволені порти, формат повідомлень та ідентифікатор контрольованого ресурсу. Підключення нових джерел повинно супроводжуватися перевіркою коректності часових позначок і відповідності отриманих даних внутрішній схемі подій.

За значної інтенсивності надходження повідомлень між модулем приймання та кореляційним обробником доцільно використовувати брокер повідомлень. Для такої конфігурації може застосовуватися RabbitMQ або інший сумісний сервіс. Його призначення полягає у тимчасовому накопиченні повідомлень та забезпеченні незалежного виконання процедур приймання й аналізу. Під час налаштування визначаються черги, правила підтвердження оброблення та обмеження повторної доставки. Додатково потрібно передбачити оброблення повідомлень, які неможливо коректно перетворити або зберегти.

Після підготовки телеметричних компонентів налаштовується модуль кореляції. До нього завантажуються правила виявлення підозрілої активності, часові інтервали аналізу та порогові значення спрацювань. Окремо визначаються характеристики захищених ресурсів, необхідні для оцінювання критичності інцидентів. Початкова конфігурація повинна забезпечувати розмежування інформаційних повідомлень, сповіщень та інцидентів. Перед увімкненням автоматизованого реагування необхідно перевірити правильність формування інцидентів на контрольованих тестових подіях.

Налаштування SOAR передбачає підключення зовнішніх засобів захисту через відповідні програмні адаптери. Для кожної інтеграції визначаються адреса API, спосіб автентифікації, допустимі операції та часові обмеження виконання запитів. Облікові дані необхідно зберігати в захищеному сховищі секретів або засобами керування конфіденційною конфігурацією. Безпосереднє розміщення паролів і токенів у вихідному коді не допускається. Для операцій, здатних

порушити доступність критичних сервісів, передбачається обов'язкове адміністративне погодження.

Рекомендована послідовність запуску системи починається з перевірки доступності мережі та сховища даних. Після цього запускається брокер повідомлень, якщо його передбачено архітектурою. Наступними активуються модуль приймання телеметрії та служба нормалізації. Після підтвердження їхньої готовності запускається кореляційний обробник і модуль керування інцидентами. Завершальним етапом активується сервіс оркестрації SOAR та програмний інтерфейс користувача. Автоматичні сценарії реагування дозволяється запускати лише після перевірки зовнішніх підключень і чинних політик доступу.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Метою розробки є дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Об'єктом дослідження є процес мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Предметом дослідження є методи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– Розроблено вітчизняний продукт мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму повинна складати одну з головних завдань роботодавця.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей виливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей виливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці

Для того, щоб об’єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв’язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов’язаних з постійною роботою за комп’ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можливе враження електричним струмом.

7.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ПЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють програмісти необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-97.

При роботі з використанням ПЕОМ відзначають наступні небезпечні та шкідливі фактори:

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території;

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м'язовий апарат;

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування засобів індивідуального захисту працівників; відповідні вимоги охорони праці, умови праці на кожному робочому місці; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання робіт; інструктаж з охорони праці; організацію контролю над станом умов праці на робочих місцях; проведення атестації робочих місць за умовами праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – одна з головних задач охорони праці.

7.3 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов’язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

7.4 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об’єкти, цілі будівлі і їх частини, поверхи, на яких відповідальний співробітник повинен проводити такі організаційні роботи.

Відповідальні особи зобов’язуються розробити, впровадити та підтримувати виконання певних інструкцій і положень на ввірених їм об’єктах. протипожежний режим відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх використання та утримання, наприклад:

– евакуаційних шляхів;

– так званих «курилок»;

– місць складування продукції та сировини;

– стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

– вентиляційного устаткування;

– засобів пожежогасіння і захисту від загорянь;

– нагрівальних приладів;

– електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об’єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об’єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань. Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4] Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» будинок можна віднести до II групи за ступенем вогнестійкості й до категорії Д за ступенем пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка поміщена в ізольований кабель. Висота проводки становить 2,2 м від рівня підлоги. Біля кожного столу

організований розподільний щит, розташований на текстолітовій пластинці, закріпленій на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемами заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежна безпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електронесправності устаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрану дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів за способом захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрої електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однією з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними силами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

7.5 Розрахункова частина

Запорукою безпечної роботи в ІТ-сфері є виконання вимог електричної безпеки, оскільки все офісне обладнання заживлюється від електричної мережі. Одним з необхідних засобів електричної безпеки є встановлення захисного заземлення.

Початкові дані, необхідні для розрахунку захисного заземлення:

– допустимий опір розповсюдженню струму в землі від заземлювального пристрою $R_{зн}$ = 10 Ом;

– питомий опір грунту в місці встановлення заземлювача $\rho_з$ = 100 Ом/м;

– тип грунту – суглинок;

– тип заземлювача – труба, діаметром d=0.058 м і довжиною l =2.1 м; – конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок проводимо за стандартною методикою [7].

Визначимо розрахунковий опір землі:

$$\rho_{рз} = \phi \cdot \rho_З$$

де ϕ – коефіцієнт сезонності, що враховує коливання питомого опору при зміні вологості ґрунту протягом року; при використанні заземлювача довжиною l = 2.1 м при глибині закладання від вершини h = 0.4 м ϕ = 1.1 для четвертої кліматичної зони.

Схема розташування заземлювачів показана на рисунку 7.1.

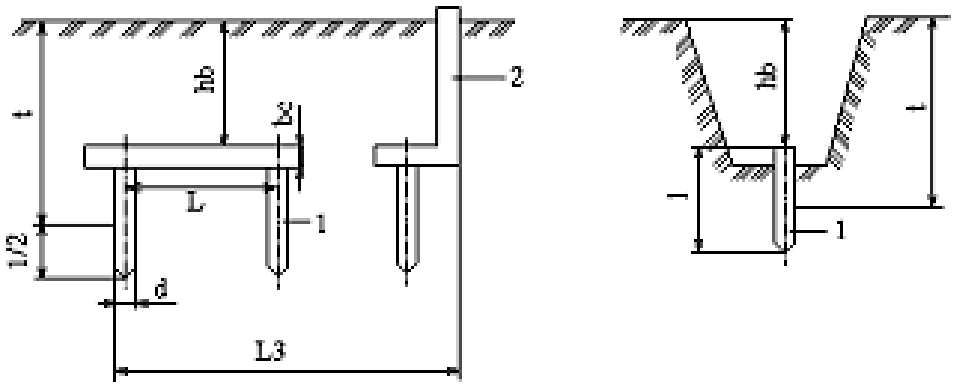

Рисунок 7.1 – Схема розташування заземлювачів

Опір землі:

$$\rho_{рз} = 1{,}1 \cdot 100 = 110 \text{ Ом·м}$$

Опір Rв, розповсюдженню струму в землі від одного вертикального заземлювача:

$$R_B = \frac{\rho_{рз}}{2\pi \cdot l}\left(\ln\frac{2 \cdot l}{d} + 0.5\ln\frac{4t + l}{4t - l}\right)$$

де

l – довжина заземлювача (l = 2.1 м);

d = 0.058 м – діаметр заземлювача при U<1 кВ та при S<100 кВА;

t – відстань від поверхні до середини заземлювача:

$$t = h + l/2 = 0.4 + 2.1/2 = 1.45 \text{ м.}$$

$$R_в = \frac{110}{2 \cdot 3.14 \cdot 2.1}\left(\ln\left(\frac{2 \cdot 2.1}{0.058}\right) + 0.5 \cdot \ln\left(\frac{4 \cdot 1.45 + 2.1}{4 \cdot 1.45 - 2.1}\right)\right) = 38.72 \text{ Ом}$$

Визначаємо потрібну кількість заземлювачів:

$$n' = \frac{R_в}{R_{зн}} = \frac{38.72}{10} = 3.9 \approx 4 \text{ шт.}$$

Коефіцієнт використання вертикальних заземлювачів враховує ефект екранування. При вибраному значенні k = a/l, де a – відстань між вертикальними заземлювачами, м; k = 1 при a = 1.5 м. Коефіцієнт використання вертикального заземлювача за довідковими даними дорівнює $\eta_В = 0,6$.

Кількість вертикальних заземлювачів з урахуванням коефіцієнту використання $\eta_В$ приблизно складає

$$n = \frac{R_в}{R_{зн} \cdot \eta_в} = \frac{38.72}{10 \cdot 0.6} = 6.45 \approx 7 \text{ шт.}$$

Довжина горизонтального заземлювача, необхідна для розміщення вертикальних заземлювачів по контуру

$$L = a \cdot n = 1.5 \cdot 7 = 10.5 \text{ м}$$

Опір горизонтального заземлювача $R_Г$, Ом, прокладеного на глибині h = 0.4 м від поверхні землі буде

$$R_г = \frac{R_{рз}}{2 \cdot 3.14 \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot h} = \frac{110}{2 \cdot 3.14 \cdot 10.5} \cdot \ln \frac{2 \cdot 10.5^2}{0.058 \cdot 0.4} = 15.7 \text{ Ом}$$

де b = 0.04 м – ширина сталевої смуги, з якої виготовлений заземлювач.

Обчислюємо загальний опір:

$$R_з = \frac{R_в \cdot R_г}{n \cdot R_г \cdot \eta_в + R_в \cdot \eta_г} = \frac{38.72 \cdot 15.7}{6 \cdot 15.7 \cdot 0.6 + 38.72 \cdot 0.34} = 8.13 \text{ Ом}$$

де $\eta_Г$ – коефіцієнт використання горизонтального заземлювача ($\eta_Г = 0.34$).

Маємо 8.13 Ом < 10 Ом, отже нормативне обмеження для потужності генераторів та трансформаторів 100 кВт і менше $R_з < R_{з.норм}$ виконується.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– Досліджена система мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

– На основі отриманих результатів досліджень створена програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності

предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рескаленко Я.Ю. Дослідження та програмна реалізація системи мережевого центру управління інформаційною безпекою (SOC) з модулями автоматизованого реагування та оркестрації інцидентів // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.
2. Massimo Bertaccini. Cryptography Algorithms. Packt Publishing. 2022. 358 p.
3. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.
4. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.
5. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
6. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
7. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
8. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
9. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p
10. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
11. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
12. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове*

наукове видання «Кібербезпека: освіта, наука, техніка», 2026, т. 2 № 34, С. 677-686.

13. Вінтенко Б.Ю., Смірнова Т.В., Миронець І.В., Смірнов О.А., Буравченко К.О. «Метод оцінки функціональної стійкості комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу АЕС». *Системи управління, навігації та зв'язку*, 2026, вип. 2(84), С. 62-69.

14. Смірнова Т.В., Усік П.С., Лисенко І.А., Буравченко К.О., Смірнов О.А. «Методологія підтримки технологічних процесів у критичній інфраструктурі з забезпеченням безпеки інформації на основі хмарних технологій». *Безпека інформації.* 2025. Том 31 № 1. С. 49-60.

15. Смірнова Т.В., Буравченко К.О., Добринчук О.А., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Хмарна технологія моніторингу ключових індикаторів ефективності технологічних процесів у критичній інфраструктурі». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. № 12(43), ч. II. С. 36-54.

16. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427.

17. Vintenko, B., Myronets, I., Smirnov, O., Smirnova, T. «Enhancing the Reliability of Input Data for Nuclear Power Plant Operator Support Systems». *International Conference on Sensors, Devices and Systems. SDaS 2025. Lecture Notes in Electrical Engineering*, 2026. vol 1617. pp 3-12/

18. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications*, 2026, pp. 587-633.

19. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

20. Chulinda L., Smirnov O., Shapenko L., Ustynova I., Bohatiuk I., Kelyp S. «The role of innovation in ensuring the safety of international civil aviation». *Ceur Workshop Proceedings*, 2025, 4024, pp. 530–542.

21. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

22. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

23. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

24. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

25. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

26. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*,

2025. vol 389. pp 377-389. Springer, Singapore.

27. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

28. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

29. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

30. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–187.

31. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

32. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

33. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

34. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic

Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

35. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

36. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

37. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

38. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings,* Volume 3504, 2023, pp. 1-11.

39. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

40. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об’єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №3(19), 2023, С. 176-196.

41. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *ІІ Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

42. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

43. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

44. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-177.

45. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings,* Volume 3187, 2022,

46. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

47. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

48. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління,*

навігації та зв'язку, 2022, № 3(69). С. 93-97.

49. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

50. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

51. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021,* Cracow, Poland, 22-25 September 2021. P. 414-418

52. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

53. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

54. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-57.*

55. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue

2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

56. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

57. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.97. No 21, 2020, P. 3334-3346.

58. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

59. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

60. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.