

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра

на тему:

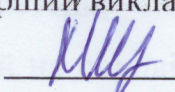
«Методи та інструменти захисту від мережних атак»

Виконав: студент 2-го курсу групи ЗКБм-24-1

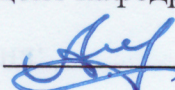
спеціальності 125 «Кібербезпека та захист інформації»

 Владислав ГУДИМЕНКО

Керівник: старший викладач кафедри КІ та КБ

 Костянтин МАТВЄЄВ

Рецензент: доцент кафедри КІ та КБ, к.т.н.

 Олексій ШЕЛУХА

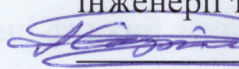
Житомир – 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека та захист інформації»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій СФІМЕНКО

6 жовтня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу магістра

Студент: Гудименко Владислав Сергійович

Тема роботи: «Методи та інструменти захисту від мережних атак»

затверджена Наказом університету від 6 жовтня 2025 р. № №566/с

Термін здачі студентом закінченої роботи 30 грудня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

1. Об'єктом дослідження є процеси моніторингу мережевого трафіку та виявлення кібератак в інформаційних системах організацій

2. Предметом дослідження є методи, алгоритми та програмні засоби виявлення мережних атак і аномалій на основі аналізу мережевого трафіку та подій безпеки.

Обладнання та програмне забезпечення:

1. Snort — система виявлення та запобігання мережевим вторгненням (IDS/IPS) з відкритим кодом, що використовує сигнатурний аналіз для виявлення відомих типів мережних атак.

2. Suricata — високопродуктивна система IDS/IPS та моніторингу мережевого трафіку з підтримкою багатопоточності, сигнатурного та аномального аналізу, а також інтеграції з іншими засобами кібербезпеки.

3. Zeek (Bro) — мережевий аналізатор безпеки, призначений для глибокого аналізу трафіку, збору метаданих та виявлення підозрілої поведінки в мережі.

4. Palo Alto Networks NGFW — міжмережевий екран нового покоління (Next-Generation Firewall), що забезпечує фільтрацію трафіку на основі додатків, користувачів і контенту, а також інтегровані функції захисту від мережевих атак.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Андрій ЄФІМЕНКО	6.10.25	ВГГ 6.10.25
2	Андрій ЄФІМЕНКО	6.10.25	ВГГ 6.10.25
3	Андрій ЄФІМЕНКО	6.10.25	ВГГ 6.10.25

Студент ВГГ Владислав ГУДИМЕНКО

Керівник МУ Костянтин МАТВЄСВ

Календарний план

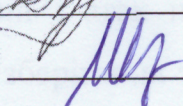
№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	06.10.2025	виконано
2	Розробка плану роботи	12.10.2025	виконано
3	Розробка розгорнутого плану	18.10.2022	виконано
4	Подача чернетки керівнику для перевірки	20.11.2022	виконано
5	Доопрацювання роботи та перевірка керівником	01.12.2025	виконано
6	Затвердження графічних матеріалів у керівника	08.12.2025	виконано
7	Проведення попереднього захисту	15.12.2025	виконано
8	Захист	29.12.2025	виконано

Студент



Владислав ГУДИМЕНКО

Керівник



Костянтин МАТВЄСВ

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕРЕЖЕВИХ ЗАГРОЗ ТА МЕТОДІВ ЗАХИСТУ	9
1.1 Класифікація мережесих атак та їх характеристики	9
1.2. Огляд існуючих систем виявлення та запобігання вторгненням	14
1.3. Аналіз інструментальних засобів моніторингу мережевої безпеки	20
1.4. Вимоги до систем захисту від мережесих атак.....	24
Висновки до розділу 1	28
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ СТАНУ МЕРЕЖІ ТА ВИЯВЛЕННЯ АТАК	29
2.1. Сценарії використання системи моніторингу	29
2.2. Архітектура системи та структура модулів захисту	34
2.3. Алгоритми виявлення мережесих аномалій та атак	42
2.4. Модель даних та структура інформаційного забезпечення	48
2.5. Об'єктна структура програмного комплексу	52
Висновки до розділу 2	58
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ТЕСТУВАННЯ СИСТЕМИ ЗАХИСТУ	60
3.1. Структура та елементи інтерфейсу застосунку.....	60
3.2. Тестування алгоритмів виявлення мережесих атак.....	64
3.3. Налаштування та встановлення системи моніторингу	71
Висновки до розділу 3	76
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83

АНОТАЦІЯ

Розробка системи моніторингу стану мережі та виявлення атак. - Кваліфікаційна робота магістра зі спеціальності 125 «Кібербезпека та захист інформації». - Державний університет «Житомирська політехніка», Житомир, 2024. - 65 с.

У першому розділі проведено аналіз сучасних мережевих загроз і методів захисту. Досліджено класифікацію мережевих атак за різними критеріями, детально розглянуто характеристики основних типів атак. Виконано огляд існуючих систем виявлення та запобігання вторгненням, проаналізовано їхні переваги та недоліки. Розглянуто інструментальні засоби моніторингу мережевої безпеки та можливості їхнього використання. Сформульовано вимоги до систем захисту від мережевих атак на основі проведеного аналізу.

У другому розділі розроблено проект системи моніторингу стану мережі та виявлення атак. Визначено основні сценарії використання системи різними категоріями користувачів. Спроектовано багатошарову архітектуру системи з визначенням структури модулів захисту та їхньої взаємодії. Розроблено алгоритми виявлення мережевих аномалій і атак на основі різних підходів. Створено модель даних для ефективного зберігання та обробки інформації про події безпеки. Спроектовано об'єктну структуру програмного комплексу з використанням сучасних принципів розробки програмного забезпечення.

У третьому розділі представлено реалізацію інтерфейсу та результати тестування системи. Детально описано структуру та основні елементи веб-інтерфейсу застосунку для різних категорій користувачів. Проведено комплексне тестування алгоритмів виявлення на різних типах атак із використанням синтетичних і реальних даних. Розроблено документацію з налаштування та встановлення системи для різних сценаріїв розгортання.

Практична цінність розробленої системи полягає в можливості її використання для захисту мереж організацій різного масштабу. Модульна архітектура дозволяє адаптувати систему до специфічних потреб, відкриті інтерфейси забезпечують інтеграцію з існуючою інфраструктурою безпеки, а комбінація різних методів виявлення забезпечує високу ефективність проти широкого спектру загроз.

Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 50 найменувань та додатків. Повний обсяг роботи становить 83 сторінок основного тексту, включаючи 5 таблиць і 7 ілюстрацій.

Ключові слова: мережева безпека, система моніторингу, виявлення атак, кібербезпека, мережеві загрози, сигнатурний аналіз, аномальна детекція, поведінковий аналіз, DDoS-атаки, SQL-ін'єкції, міжсайтовий скриптинг, архітектура безпеки, модель даних, веб-інтерфейс, тестування продуктивності.

ANNOTATION

Development of a Network Monitoring and Attack Detection System. - Master's thesis in specialty 125 "Cybersecurity and Information Protection". - Zhytomyr Polytechnic State University, Zhytomyr, 2024. - 65 p.

The first chapter analyzes modern network threats and protection methods. The classification of network attacks according to various criteria is examined, and the characteristics of main types of attacks are considered in detail. A review of existing intrusion detection and prevention systems is performed, analyzing their advantages and disadvantages. Network security monitoring tools and possibilities of their application are reviewed. Requirements for network attack protection systems are formulated based on the conducted analysis.

The second chapter presents the design of a network monitoring and attack detection system. Main use case scenarios for different user categories are defined. A multi-layered system architecture is designed with definition of protection module structure and their interaction. Algorithms for detecting network anomalies and attacks based on different approaches are developed. A data model for efficient storage and processing of security event information is created. The object structure of the software complex is designed using modern software development principles.

The third chapter presents the interface implementation and system testing results. The structure and main elements of the application's web interface for different user categories are described in detail. Comprehensive testing of detection algorithms on various attack types using synthetic and real data is conducted. Documentation for system configuration and installation for different deployment scenarios is developed.

The practical value of the developed system lies in its applicability for protecting networks of organizations of various scales. The modular architecture allows adapting the system to specific needs, open interfaces ensure integration with

existing security infrastructure, and the combination of different detection methods provides high efficiency against a wide spectrum of threats.

The work consists of an introduction, three chapters, conclusions, a list of references containing 50 items, and appendices. The total volume of the work is 92 pages of main text, including 5 tables and 7 illustrations.

Keywords: network security, monitoring system, attack detection, cybersecurity, network threats, signature analysis, anomaly detection, behavioral analysis, DDoS attacks, SQL injections, cross-site scripting, security architecture, data model, web interface, performance testing.

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням кількості та складності кіберзагроз, що становлять серйозну небезпеку для інформаційних систем організацій усіх рівнів. За даними аналітичних агентств, кількість кібератак щорічно зростає на 30-40 відсотків, а їхня складність постійно еволюціонує з появою нових методів обходу систем захисту. Фінансові втрати від кіберінцидентів досягають мільярдів доларів щорічно, що робить питання мережевої безпеки критично важливим для бізнесу та державних організацій.

Актуальність теми дослідження обумовлена необхідністю створення ефективних систем моніторингу та виявлення мережевих атак, здатних протистояти сучасним загрозам. Традиційні підходи до забезпечення безпеки, що базуються виключно на периметральному захисті через міжмережеві екрани, виявляються недостатніми для протидії складним цільовим атакам, внутрішнім загрозам, атакам нульового дня. Організації потребують комплексних систем моніторингу, що забезпечують виявлення загроз у режимі реального часу, аналіз великих обсягів мережевого трафіку, кореляцію подій з різних джерел для виявлення складних багатоетапних атак.

Метою роботи є дослідження методів виявлення мережевих загроз та розробка системи моніторингу стану мережі з можливістю ефективного виявлення широкого спектру мережевих атак.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Провести аналіз проблематики мережевої безпеки, методів виявлення загроз та засобів вирішення проблеми захисту мереж;
- Розробити архітектуру системи моніторингу, модель даних та інші підготовчі проєктувальні рішення для побудови системи;

- Реалізувати систему моніторингу мережевої безпеки з використанням обраних технологій та підходів;
- Провести аналіз отриманих результатів та оцінити ефективність розробленої системи.

Об'єктом дослідження є процеси моніторингу мережевого трафіку та виявлення кібератак в інформаційних системах організацій.

Предметом дослідження є методи, алгоритми та програмні засоби виявлення мережевих атак і аномалій на основі аналізу мережевого трафіку та подій безпеки.

Методи дослідження. У роботі використовувалися методи системного аналізу для дослідження існуючих рішень і формування вимог до системи, методи об'єктно-орієнтованого проектування для розробки архітектури програмного комплексу, алгоритми обробки мережевих пакетів для захоплення та аналізу трафіку, методи статистичного аналізу для виявлення аномалій у мережевій активності, алгоритми співставлення з патернами для сигнатурного виявлення загроз, методи машинного навчання для поведінкового аналізу, методи тестування програмного забезпечення для валідації коректності роботи системи.

Практичне значення отриманих результатів полягає в можливості використання розробленої системи для моніторингу мережевої безпеки в організаціях різного масштабу. Система забезпечує виявлення широкого спектру мережевих атак у режимі реального часу, що дозволяє своєчасно реагувати на загрози та мінімізувати потенційні збитки. Модульна архітектура та відкриті інтерфейси дозволяють інтегрувати систему з існуючою інфраструктурою безпеки організації. Зручний веб-інтерфейс знижує поріг входу для адміністраторів і прискорює процес розслідування інцидентів. Результати тестування підтверджують конкурентну точність виявлення та продуктивність порівняно з існуючими рішеннями.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 50 найменувань та додатків. Повний обсяг роботи становить 92 сторінок основного тексту, включаючи 5 таблиць і 7 ілюстрацій.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕРЕЖЕВИХ ЗАГРОЗ ТА МЕТОДІВ ЗАХИСТУ

1.1 Класифікація мережесих атак та їх характеристики

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням кількості та складності кіберзагроз, що становлять серйозну небезпеку для інформаційних систем усіх рівнів. Мережесі атаки еволюціонують, адаптуючись до нових засобів захисту та використовуючи інноваційні методи проникнення в захищені системи. Розуміння природи цих загроз є фундаментальною основою для побудови ефективних систем кібербезпеки. Мережесі атаки можна систематизувати за різними критеріями, що дозволяє створити комплексну класифікацію для подальшого аналізу та розробки методів протидії. Основними класифікаційними ознаками є технічний рівень реалізації атаки, цільова спрямованість дій злоумисника, використовувані методи та інструменти, а також потенційні наслідки для цільової системи.

За рівнем моделі OSI мережесі атаки розподіляються відповідно до семирівневої архітектури мережесих протоколів. На фізичному та каналному рівнях злоумисники можуть здійснювати атаки через фізичне втручання в мережесу інфраструктуру або експлуатацію вразливостей протоколів локальних мереж. Типовими прикладами є атаки MAC-flooding, коли таблиця комутатора переповнюється фіктивними записами, що призводить до його переходу в режим концентратора, та ARP-spoofing, що дозволяє перехоплювати трафік між пристроями локальної мережі шляхом підробки ARP-повідомлень.

Таблиця 1.1

Класифікація мережесих атак за рівнями OSI

Рівень OSI	Типи атак
Мережесий	IP-спуфінг, ICMP-флуд, фрагментаційні атаки
Транспортний	SYN-flood, UDP-flood, атаки на TCP з'єднання

Прикладний	HTTP-flood, SQL-ін'єкції, XSS, CSRF, атаки на веб-застосунки
------------	--

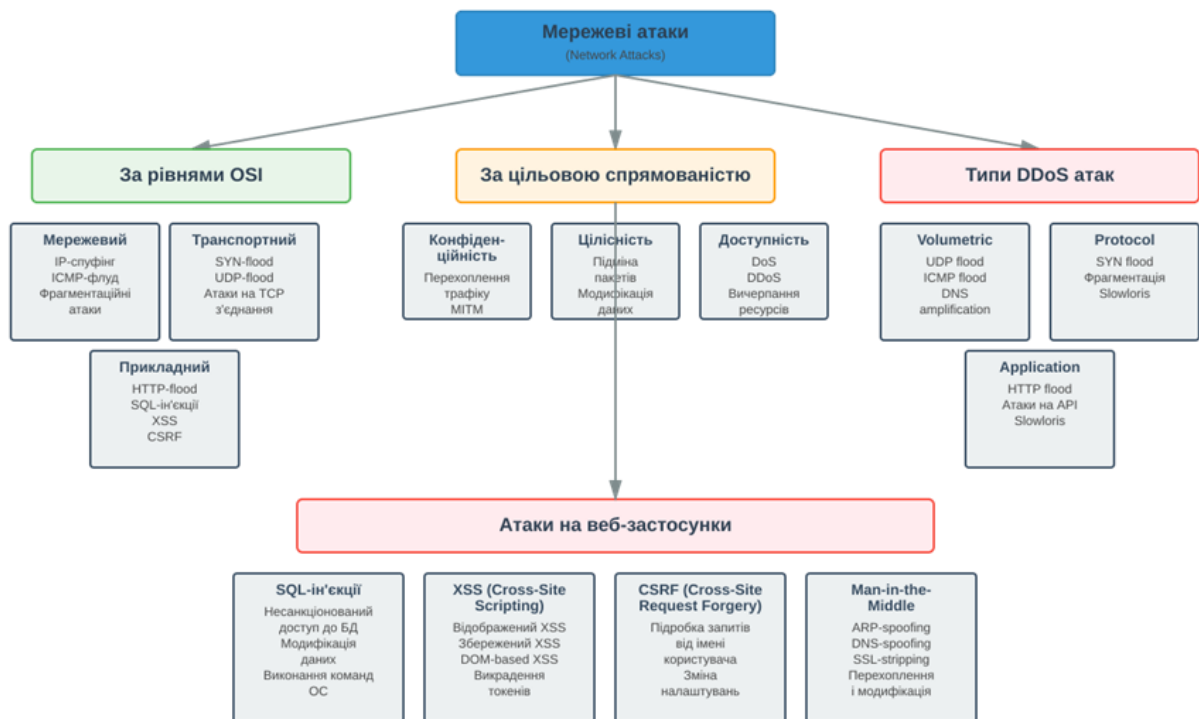


Рисунок 1.1 – Класифікація мережевих атак

Мережевий рівень стає ціллю для атак, що використовують недоліки протоколу IP та маршрутизації. IP-spoofing передбачає підміну адреси відправника пакетів для приховування джерела атаки або обходу систем фільтрації. Фрагментаційні атаки експлуатують механізм розділення IP-пакетів на фрагменти, створюючи спеціально сформовані послідовності, що можуть призвести до відмови в обслуговуванні або обходу систем безпеки. Транспортний рівень характеризується атаками на протоколи TCP та UDP. SYN-flood атаки використовують механізм встановлення TCP-з'єднання, надсилаючи велику кількість запитів на встановлення з'єднання без завершення процесу, що призводить до вичерпання ресурсів сервера. UDP-flood атаки базуються на

безз'єднувальній природі протоколу UDP, генеруючи масивні потоки даних на довільні порти цільової системи [1-2].

Прикладний рівень залишається найбільш вразливим, оскільки саме тут реалізується безпосередня взаємодія з користувачами та обробка даних. Атаки на веб-застосунки включають SQL-ін'єкції, міжсайтовий скриптинг, підробку міжсайтових запитів та інші методи експлуатації вразливостей програмного коду. Ці атаки можуть призводити до несанкціонованого доступу до баз даних, викрадення облікових записів користувачів або виконання довільного коду на стороні сервера чи клієнта.

За цільовою спрямованістю мережеві атаки поділяються на три основні категорії відповідно до класичної тріади інформаційної безпеки. Атаки на конфіденційність спрямовані на несанкціонований доступ до інформації та включають перехоплення мережевого трафіку, атаки типу "людина посередині", витік даних через вразливості застосунків. Сучасні методи шифрування значно ускладнюють перехоплення даних, але зловмисники розробляють нові підходи, включаючи атаки на криптографічні протоколи та використання соціальної інженерії для отримання ключів дешифрування. Атаки на цілісність мають на меті несанкціоноване модифікування даних під час їх передачі або зберігання. Підміна пакетів у мережі, модифікація контенту веб-сторінок, втручання в роботу баз даних є типовими проявами таких атак. Особливу небезпеку становлять атаки, що поєднують порушення конфіденційності та цілісності, дозволяючи не лише читати, але й змінювати дані без виявлення.

Атаки на доступність намагаються зробити інформаційні ресурси недоступними для легітимних користувачів. Відмова в обслуговуванні може досягатися різними методами, від простого перевантаження мережевих каналів до складних багатовекторних атак, що одночасно впливають на різні компоненти інфраструктури. Ці атаки становлять особливу загрозу для критично

важливих систем, де навіть короточасна недоступність може призвести до значних фінансових втрат або порушення безпеки [3-4].

Атаки типу DoS та DDoS представляють окрему категорію загроз через їх поширеність та потенційний масштаб впливу. Простий DoS здійснюється з одного джерела та базується на вичерпанні певного ресурсу цільової системи. Це може бути процесорний час, оперативна пам'ять, мережева пропускна здатність або кількість одночасних з'єднань. Сучасні системи зазвичай здатні протистояти простим DoS атакам завдяки достатній потужності обладнання та базовим механізмам захисту.

Розподілений DoS кардинально змінює ситуацію, використовуючи множину скомпрометованих комп'ютерів для одночасної атаки на ціль. Ботнети, що об'єднують тисячі або навіть мільйони заражених пристроїв, можуть генерувати трафік обсягом в терабіти на секунду, що перевищує можливості більшості систем захисту. Координація атаки через централізовану або децентралізовану систему управління робить DDoS надзвичайно ефективною зброєю кіберзлочинців. Volumetric атаки фокусуються на насиченні мережевих каналів величезним обсягом трафіку. UDP flood генерує масивні потоки UDP-пакетів на довільні порти, змушуючи цільову систему витрачати ресурси на обробку та генерацію ICMP-повідомлень про недоступність порту. ICMP flood використовує протокол службових повідомлень для перевантаження мережі. DNS amplification застосовує відкриті DNS-сервери як підсилювачі, надсилаючи невеликі запити з підробленою адресою відправника та отримуючи значно більші відповіді, спрямовані на жертву [5-7].

Protocol атаки експлуатують особливості та вразливості мережевих протоколів. SYN flood є класичним прикладом, створюючи величезну кількість напіввідкритих з'єднань та вичерпуючи таблицю станів на сервері або міжмережевому екрані. Фрагментаційні атаки використовують механізми розбиття пакетів для обходу систем безпеки або спричинення аномальної

поведінки при збиранні фрагментів. Slowloris підтримує багато з'єднань відкритими якомога довше, надсилаючи часткові HTTP-запити та не завершуючи їх.

Application layer атаки є найскладнішими для виявлення, оскільки генерують трафік, що нагадує легітимні запити користувачів. HTTP flood імітує нормальну активність веб-переглядача, але з інтенсивністю, що перевищує можливості сервера. Атаки на API використовують ресурсомісткі запити до програмних інтерфейсів, вичерпуючи потужність серверів додатків та баз даних. Такі атаки потребують глибокого розуміння роботи цільового застосунку та є складними для автоматичного виявлення. Атаки на веб-застосунки становлять окрему категорію через їх різноманітність та критичність наслідків. SQL-ін'єкції використовують недостатню валідацію вхідних даних для виконання довільних запитів до бази даних. Зловмисник може отримати доступ до всієї інформації, модифікувати або видалити дані, а в деяких випадках навіть виконати команди операційної системи. Автоматизовані інструменти сканування дозволяють швидко виявляти вразливі застосунки, що робить цей тип атак надзвичайно поширеним.

Міжсайтовий скриптинг дозволяє впроваджувати шкідливий JavaScript-код на веб-сторінки, що виконується в браузері жертви. Відображений XSS передає зловмисний код через параметри URL або форми, збережений XSS зберігає код в базі даних застосунку, DOM-based XSS маніпулює об'єктною моделлю документа на стороні клієнта. Наслідки можуть включати викрадення сесійних токенів, перенаправлення користувачів на фішингові сайти або виконання дій від імені жертви.

Підробка міжсайтових запитів змушує браузер автентифікованого користувача виконувати небажані дії на довіреному сайті. Атака базується на тому, що браузер автоматично додає cookies до запитів, дозволяючи зловмисникові ініціювати операції від імені жертви. Це може призвести до зміни

налаштувань облікового запису, здійснення транзакцій або розкриття конфіденційної інформації.

Атаки типу "людина посередині" представляють складну категорію загроз, що поєднують елементи порушення конфіденційності та цілісності. Зловмисник позиціонує себе між двома комунікуючими сторонами, перехоплюючи та потенційно модифікуючи весь трафік. В локальних мережах це часто досягається через ARP-spoofing, що дозволяє переспрямувати трафік через систему атакуючого. На рівні DNS підrobка відповідей дозволяє переспрямовувати користувачів на контрольовані зловмисником сервери.

SSL-stripping є особливо підступною формою MITM атаки, що знижує захищене HTTPS з'єднання до незахищеного HTTP. Зловмисник перехоплює спроби встановлення захищеного з'єднання та змінює HTTPS посилання на HTTP, встановлюючи власне захищене з'єднання з сервером, але надаючи користувачеві незахищене. Це дозволяє читати весь трафік, включаючи паролі та конфіденційні дані, без викликання попереджень браузера.

Атаки на бездротові мережі експлуатують специфічні вразливості технологій Wi-Fi. Деавтентифікаційні атаки примусово роз'єднують клієнтів з точкою доступу, що може використовуватися для відмови в обслуговуванні або перехоплення процесу повторної автентифікації. Evil twin атаки створюють фальшиву точку доступу з іменем легітимної мережі, змушуючи пристрої підключатися до неї та розкриваючи весь трафік. Крекінг паролів WPA/WPA2 через перехоплення handshake та офлайн-атаки залишається реальною загрозою при використанні слабких паролів.

1.2. Огляд існуючих систем виявлення та запобігання вторгненням

Системи виявлення та запобігання вторгненням є критичними компонентами сучасної архітектури кібербезпеки, що забезпечують моніторинг мережевого трафіку та активності систем для ідентифікації потенційних загроз.

Ці системи еволюціонували від простих сигнатурних детекторів до складних платформ, що використовують машинне навчання та поведінковий аналіз для виявлення відомих та невідомих загроз.

Системи виявлення вторгнень поділяються на дві основні категорії відповідно до джерела аналізованих даних. Мережеві системи виявлення вторгнень здійснюють моніторинг мережевого трафіку, аналізуючи пакети даних на предмет підозрілої активності. Вони розміщуються в стратегічних точках мережі та працюють в пасивному режимі, не впливаючи на потік трафіку. Хостові системи виявлення вторгнень встановлюються на окремих серверах або робочих станціях, моніторячи системні виклики, журнали подій, зміни файлової системи та активність процесів.

Методи виявлення загроз в сучасних системах базуються на різних підходах з власними перевагами та обмеженнями. Сигнатурне виявлення використовує базу даних відомих патернів атак та шукає співпадіння в моніторимому трафіку або активності. Цей метод є надзвичайно ефективним для виявлення відомих загроз з мінімальною кількістю хибних спрацьовувань, але безсилий проти нових, невідомих атак. Регулярне оновлення бази сигнатур є критично важливим для підтримання ефективності такого підходу.

Аномальне виявлення базується на створенні моделі нормальної поведінки системи та ідентифікації відхилень від цієї моделі як потенційних загроз. Статистичний аномальний аналіз використовує математичні методи для визначення профілю нормальної активності та встановлення порогів для детекції аномалій [8]. Поведінковий аналіз вивчає типові дії користувачів та процесів, виявляючи незвичну активність, що може вказувати на компрометацію. Основною перевагою цього підходу є здатність виявляти невідомі загрози, але ціною вищого рівня хибних спрацьовувань. Snort є однією з найпопулярніших відкритих систем виявлення вторгнень, що поєднує можливості сигнатурного виявлення з елементами аномального аналізу. Система

використовує гнучку мову правил для опису патернів атак, дозволяючи адміністраторам створювати власні сигнатури для специфічних загроз. Архітектура Snort включає препроцесори, що виконують нормалізацію трафіку та виявлення аномалій протоколів, та механізм виявлення, що співставляє пакети з правилами. Модульна будова дозволяє розширювати функціональність через плагіни та інтеграцію з іншими інструментами безпеки.

Suricata представляє наступне покоління систем виявлення вторгнень, розроблене з урахуванням сучасних вимог до продуктивності та функціональності. Багатопоточна архітектура дозволяє ефективно використовувати багатоядерні процесори, забезпечуючи аналіз високошвидкісного трафіку без втрати пакетів. Система підтримує сумісність з правилами Snort, але додає власні розширення для глибокого аналізу протоколів та контенту. Вбудована підтримка скриптів Lua дозволяє створювати складну логіку виявлення, що враховує контекст комунікації.

Zeek, раніше відомий як Bro, займає унікальну нішу серед систем виявлення вторгнень завдяки орієнтації на аналіз мережевої активності високого рівня. Замість простого співставлення з сигнатурами, Zeek будує детальне представлення мережевих комунікацій, генеруючи журнали про всі з'єднання, використані протоколи та передані дані. Мова скриптів дозволяє аналітикам створювати складні правила виявлення, що базуються на поведінці мережі протягом часу. Така архітектура робить Zeek незамінним інструментом для поглибленого аналізу мережевої безпеки та розслідування інцидентів [9].

Таблиця 1.2

Порівняльна характеристика систем виявлення вторгнень

Система	Метод виявлення	Продуктивність	Ліцензія
Snort	Сигнатурний	До 1 Гбіт/с	Open Source
Suricata	Сигнатурний +	До 10 Гбіт/с	Open Source

	аномалії		
Zeek	Аналіз протоколів	До 10 Гбіт/с	Open Source
Palo Alto NGFW	Комбінований	До 100 Гбіт/с	Комерційна

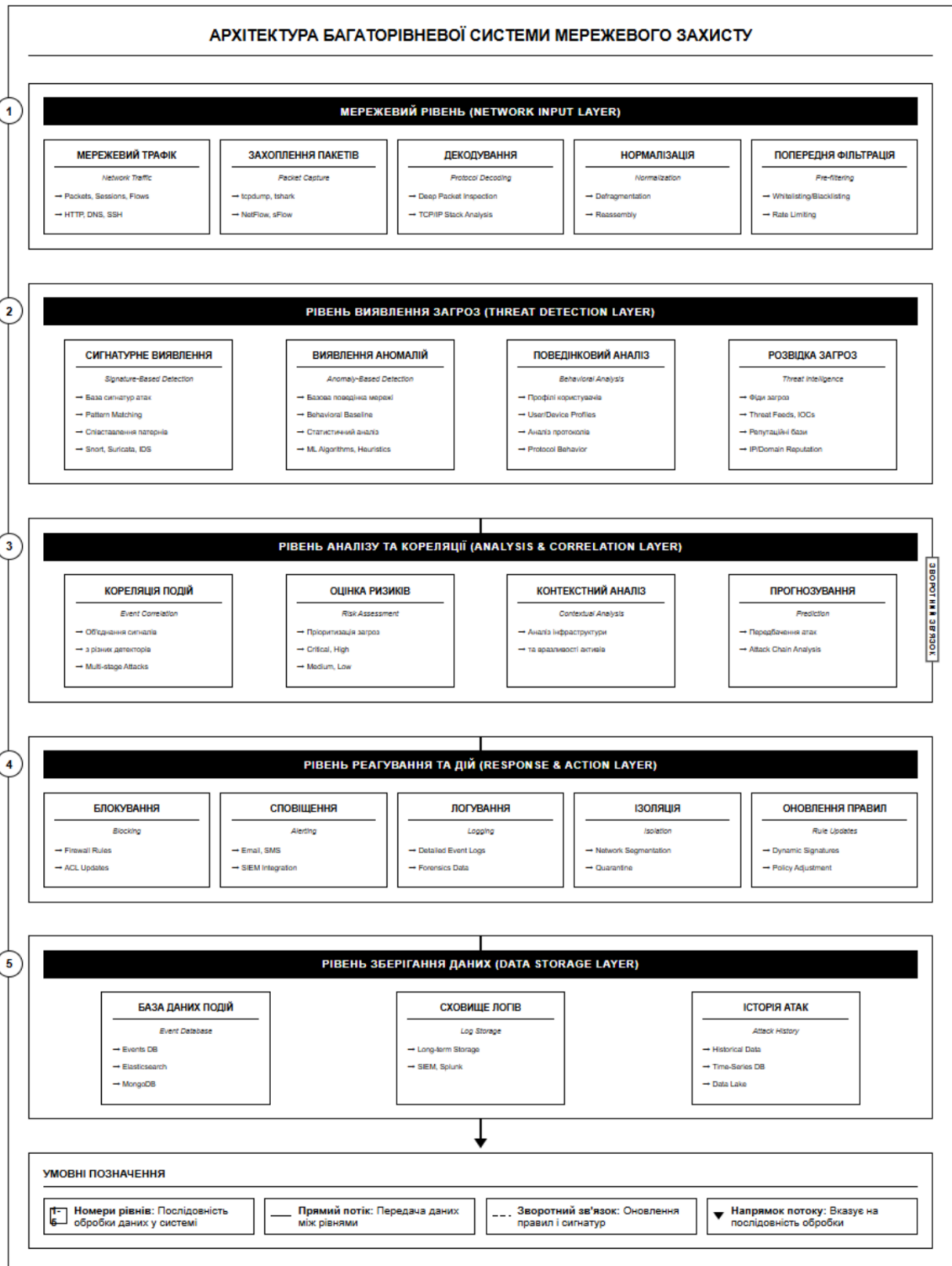


Рисунок 1.2 – Архітектура системи виявлення та запобігання вторгненням

Системи запобігання вторгненням розширюють концепцію виявлення, додаючи можливість активного блокування виявлених загроз. На відміну від пасивних систем виявлення, системи запобігання розміщуються безпосередньо в шляху мережевого трафіку, аналізуючи кожен пакет перед його пересиланням. Виявлення підозрілої активності призводить до блокування відповідних пакетів або з'єднань, запобігаючи реалізації атаки. Це потребує високої точності виявлення, оскільки хибні спрацьовування можуть блокувати легітимний трафік та порушувати роботу сервісів.

Palo Alto Networks Next-Generation Firewall є прикладом комерційного рішення, що поєднує функції міжмережевого екрану та системи запобігання вторгненням. Платформа використовує глибоку інспекцію пакетів для ідентифікації застосунків незалежно від портів та протоколів, дозволяючи створювати політики безпеки на рівні застосунків. Інтегрована система запобігання вторгненням забезпечує захист від відомих та невідомих загроз, використовуючи комбінацію сигнатурного виявлення, евристичного аналізу та глобальної загрозової розвідки.

Cisco Firepower представляє комплексне рішення для захисту мережі, що інтегрує технологію Snort з додатковими можливостями аналізу та реагування. Система здійснює контроль застосунків, виявлення та блокування зловмисного програмного забезпечення, глибоку інспекцію SSL-трафіку. Інтеграція з платформою загрозової розвідки Talos забезпечує доступ до актуальної інформації про нові загрози та індикатори компрометації [10-11].

Відкриті рішення, такі як Security Onion, об'єднують кілька інструментів виявлення в єдину платформу, що спрощує розгортання та управління системою моніторингу безпеки. Дистрибутив включає Suricata для виявлення вторгнень, Zeek для аналізу мережевих протоколів, Elasticsearch для зберігання та індексації даних, Kibana для візуалізації. Інтеграція цих компонентів створює

потужну систему, доступну організаціям з обмеженими бюджетами на кібербезпеку.

Сучасні системи виявлення все частіше інкорпорують технології машинного навчання для покращення точності виявлення та зменшення кількості хибних спрацьовувань. Алгоритми навчаються на історичних даних про нормальну та зловмисну активність, створюючи моделі, здатні виявляти складні патерни, що не піддаються опису простими правилами. Глибоке навчання дозволяє автоматично виділяти значущі ознаки з великих обсягів даних, покращуючи виявлення невідомих загроз. Однак ці підходи потребують значних обчислювальних ресурсів та якісних даних для навчання.

Виклики при впровадженні систем виявлення та запобігання вторгненням включають балансування між чутливістю виявлення та кількістю хибних спрацьовувань. Надто агресивні налаштування призводять до великої кількості помилкових тривог, що перевантажують команди безпеки та знижують довіру до системи. Недостатня чутливість дозволяє загрозам залишатися непоміченими. Масштабування є іншим значним викликом, оскільки зростання обсягів трафіку вимагає пропорційного збільшення обчислювальних потужностей [12-13].

1.3. Аналіз інструментальних засобів моніторингу мережевої безпеки

Ефективний моніторинг мережевої безпеки вимагає комплексного підходу та використання різноманітних інструментів, кожен з яких вирішує специфічні завдання збору, аналізу та візуалізації інформації про стан безпеки. Сучасний ландшафт інструментів включає рішення для аналізу трафіку, моніторингу подій безпеки, оцінки вразливостей та координації реагування на інциденти.

Засоби аналізу мережевого трафіку забезпечують детальний огляд комунікацій в мережі, дозволяючи виявляти аномалії та розслідувати інциденти. Wireshark є універсальним інструментом для захоплення та інтерактивного аналізу мережевих пакетів. Графічний інтерфейс дозволяє детально

досліджувати кожен пакет, відстежувати потоки даних, аналізувати протоколи всіх рівнів. Потужна система фільтрів забезпечує швидкий пошук потрібної інформації в великих обсягах захопленого трафіку. Хоча Wireshark орієнтований на інтерактивний аналіз, його консольна версія tshark дозволяє автоматизувати обробку захоплених даних [14].

tcpdump представляє класичний консольний інструмент для захоплення трафіку, що залишається незамінним для швидкого аналізу на серверах без графічного інтерфейсу. Компактність та ефективність роблять його ідеальним для використання в скриптах автоматизації. Можливість зберігання захопленого трафіку в стандартному форматі pcap забезпечує сумісність з іншими інструментами аналізу. Гнучкі фільтри дозволяють захоплювати лише потрібний трафік, зменшуючи навантаження на систему та обсяг даних для зберігання.

NetworkMiner спеціалізується на форензичному аналізі мережевого трафіку, автоматично виділяючи та класифікуючи об'єкти з захоплених пакетів. Система може ідентифікувати хости в мережі, використовувани операційні системи, передані файли, облікові дані, сесії комунікацій. Пасивний режим роботи дозволяє здійснювати моніторинг без активного сканування, що є важливим для дотримання політик безпеки. Можливість аналізу PCAP файлів робить інструмент цінним для розслідування інцидентів безпеки.

Системи управління інформацією та подіями безпеки агрегують дані з множини джерел, забезпечуючи централізований моніторинг та кореляцію подій. Splunk є однією з найпотужніших платформ для збору, індексації та аналізу машинних даних. Здатність обробляти величезні обсяги логів з різноманітних джерел робить Splunk універсальним інструментом для моніторингу безпеки великих інфраструктур. Гнучка мова пошуку SPL дозволяє створювати складні запити для виявлення патернів та аномалій. Візуалізація даних через дашборди забезпечує оперативний огляд стану безпеки [15-16].

Elastic Stack, що включає Elasticsearch, Logstash та Kibana, представляє відкриту альтернативу комерційним SIEM рішенням. Logstash забезпечує збір та обробку логів з різних джерел, Elasticsearch здійснює індексацію та зберігання даних, Kibana надає інтерфейс для пошуку та візуалізації. Гнучкість та масштабованість робить цей стек популярним вибором для організацій різного розміру. Додавання Beats як агентів збору даних розширює можливості моніторингу різноманітних компонентів інфраструктури.

Graylog фокусується на управлінні логами з акцентом на простоту налаштування та використання. Централізоване збирання логів з мережевих пристроїв, серверів, застосунків забезпечує повний огляд подій в інфраструктурі. Система потоків дозволяє маршрутизувати різні типи логів до відповідних індексів, оптимізуючи зберігання та пошук. Вбудовані механізми оповіщення інформують адміністраторів про критичні події в реальному часі.

Інструменти оцінки вразливостей допомагають проактивно виявляти слабкі місця в захисті до того, як ними скористаються зловмисники. Nessus є комплексним сканером вразливостей, що підтримує величезну базу перевірок для мережевих сервісів, операційних систем, застосунків. Регулярне сканування інфраструктури дозволяє виявляти нові вразливості, що з'являються через виявлення недоліків в програмному забезпеченні або неправильні конфігурації. Детальні звіти надають інформацію про виявлені проблеми та рекомендації щодо їх усунення [17-18].

OpenVAS представляє відкриту альтернативу комерційним сканерам вразливостей, що активно розвивається спільнотою. Регулярні оновлення бази вразливостей забезпечують актуальність перевірок. Можливість інтеграції в автоматизовані процеси дозволяє включати оцінку вразливостей в пайплайни безперервної інтеграції та доставки. Веб-інтерфейс спрощує управління завданнями сканування та аналіз результатів.

Nmap залишається незамінним інструментом для дослідження мережі та інвентаризації активів. Виявлення активних хостів, відкритих портів, використовуваних сервісів та версій програмного забезпечення надає фундаментальну інформацію про структуру мережі. Скриптовий движок NSE розширює можливості Nmap, дозволяючи виконувати складні перевірки безпеки, виявлення вразливостей, збір додаткової інформації про цілі. Гнучкість та потужність роблять Nmap стандартним інструментом для аудиту безпеки мережі.

Засоби тестування на проникнення моделюють дії злоумисників для оцінки ефективності захисту. Metasploit Framework є комплексною платформою для розробки та виконання експлойтів, що включає величезну базу відомих вразливостей та інструменти для пост-експлуатації. Модульна архітектура дозволяє легко додавати нові експлойти та розширювати функціональність. Інтеграція з іншими інструментами безпеки створює потужне середовище для всебічного тестування захисту [19-20].

Burp Suite спеціалізується на тестуванні безпеки веб-застосунків, надаючи інструменти для перехоплення та модифікації HTTP-трафіку, автоматизованого сканування вразливостей, тестування автентифікації та авторизації. Проксі-сервер дозволяє аналітику повністю контролювати комунікацію між браузером та веб-застосунком, виявляючи вразливості через маніпуляцію запитами та відповідями. Розширення через плагіни адаптує інструмент до специфічних потреб тестування.

Платформи threat intelligence забезпечують доступ до актуальної інформації про загрози, індикатори компрометації, тактики злоумисників. MISP є відкритою платформою для обміну інформацією про загрози, що дозволяє організаціям колективно покращувати розуміння ландшафту загроз. Структуроване зберігання індикаторів компрометації забезпечує їх автоматичне

використання в системах захисту. Таксономії та галактики класифікують загрози відповідно до усталених фреймворків.

OpenCTI об'єднує інформацію з різних джерел threat intelligence, створюючи єдине представлення про загрози та їх взаємозв'язки. Графова база даних дозволяє моделювати складні відношення між загрозами, акторами, вразливостями, індикаторами. Автоматичне збагачення даних через конектори до зовнішніх джерел покращує якість інформації. Візуалізація графу знань полегшує розуміння контексту загроз [21].

Інтеграція різних інструментів моніторингу створює комплексну систему виявлення та реагування на загрози. Використання форматів обміну даними, таких як STIX та TAXII, забезпечує сумісність між різними рішеннями. Автоматизація через оркестрацію SOAR платформ координує дії різних інструментів, прискорюючи реагування на інциденти.

1.4. Вимоги до систем захисту від мережевих атак

Розробка ефективної системи захисту від мережевих атак вимагає чіткого визначення функціональних та нефункціональних вимог, що забезпечують адекватний рівень безпеки при збереженні прийнятної продуктивності та зручності використання. Ці вимоги формуються на основі аналізу загроз, регуляторних вимог, специфіки захищуваної інфраструктури та доступних ресурсів.

Функціональні вимоги визначають що система повинна робити для забезпечення захисту. Виявлення атак є основною функцією, що вимагає здатності ідентифікувати широкий спектр загроз через сигнатурний аналіз відомих патернів атак та аномальне виявлення невідомих загроз. Система повинна підтримувати регулярне оновлення бази сигнатур для забезпечення захисту від нових загроз. Можливість створення власних правил виявлення дозволяє адаптувати систему до специфічних потреб організації.

Моніторинг мережевого трафіку повинен здійснюватися в реальному часі з мінімальною затримкою між виникненням події та її виявленням. Система має аналізувати трафік на всіх рівнях моделі OSI, від низькорівневих мережевих протоколів до застосунків. Глибока інспекція пакетів забезпечує виявлення загроз, що приховуються в легітимному трафіку. Реконструкція сесій та потоків даних дозволяє виявляти атаки, що розподілені в часі або розбиті на множину пакетів.

Журналювання та аудит є критично важливими для розслідування інцидентів та дотримання регуляторних вимог. Система повинна зберігати детальну інформацію про всі виявлені події безпеки, включаючи часові мітки, джерела та призначення, тип загрози, вжиті дії. Цілісність журналів має гарантуватися криптографічними методами для запобігання маніпуляціям. Ротація та архівування логів забезпечує довгострокове зберігання при оптимальному використанні ресурсів.

Оповіщення про критичні події безпеки повинно здійснюватися в реальному часі через множину каналів, включаючи електронну пошту, SMS, інтеграцію з системами управління інцидентами. Інтелектуальна фільтрація та пріоритезація оповіщень запобігає перевантаженню адміністраторів великою кількістю тривог низької важливості. Ескалація некритичних оповіщень при їх ігноруванні гарантує увагу до потенційних проблем.

Візуалізація стану безпеки через дашборди надає оперативний огляд ситуації, дозволяючи швидко оцінювати наявність активних загроз та ефективність захисту. Графічне представлення трендів та статистики допомагає виявляти закономірності та аномалії. Деталізація дозволяє переходити від загального огляду до детального аналізу конкретних подій.

Інтеграція з іншими системами безпеки створює єдину екосистему захисту. Взаємодія з міжмережевими екранами дозволяє автоматично блокувати джерела атак. Обмін інформацією з SIEM системами забезпечує кореляцію

подій з різних джерел. Використання threat intelligence збагачує контекст виявлених загроз, покращуючи розуміння їх природи та потенційних наслідків.

Нефункціональні вимоги визначають якісні характеристики системи, що впливають на її придатність для експлуатації. Продуктивність є критичним параметром, оскільки система не повинна створювати вузькі місця в мережевій інфраструктурі. Пропускна здатність аналізу трафіку має відповідати або перевищувати швидкість мережеских каналів. Затримка обробки пакетів повинна бути мінімальною для запобігання впливу на користувацький досвід. Ефективне використання обчислювальних ресурсів дозволяє розгортати систему на доступному обладнанні.

Масштабованість забезпечує адаптацію системи до зростання інфраструктури. Горизонтальне масштабування через додавання сенсорів моніторингу дозволяє покривати розподілені мережі. Вертикальне масштабування забезпечує обробку збільшених обсягів трафіку. Централізоване управління множиною екземплярів системи спрощує адміністрування великих розгортань.

Надійність системи захисту є фундаментальною вимогою, оскільки її відмова створює вікно вразливості. Відмовостійка архітектура з резервуванням критичних компонентів забезпечує безперервність роботи при збоях окремих елементів. Автоматичне відновлення після збоїв мінімізує час недоступності. Регулярне резервне копіювання конфігурацій та даних дозволяє швидко відновити функціональність.

Безпека самої системи захисту є парадоксальною, але критично важливою вимогою. Система не повинна створювати нових вразливостей в інфраструктурі. Захист управлінських інтерфейсів через сильну автентифікацію та шифрування запобігає несанкціонованому доступу. Регулярне оновлення програмного забезпечення усуває виявлені вразливості. Мінімізація поверхні атак через вимкнення непотрібних сервісів зменшує ризики компрометації.

Зручність використання впливає на ефективність експлуатації системи. Інтуїтивний інтерфейс знижує поріг входу для адміністраторів. Якісна документація та навчальні матеріали прискорюють освоєння функціональності. Автоматизація рутинних операцій звільняє час фахівців для аналізу та реагування на складні загрози. Гнучкі можливості налаштування дозволяють адаптувати систему до специфічних потреб.

Сумісність з існуючою інфраструктурою мінімізує витрати на впровадження. Підтримка стандартних протоколів та форматів даних забезпечує інтеграцію з різноманітними системами. Відкриті API дозволяють розробляти власні інтеграції та розширення. Незалежність від конкретних постачальників обладнання забезпечує гнучкість архітектурних рішень.

Економічна ефективність балансує вартість впровадження та експлуатації з рівнем забезпечуваного захисту. Відкриті рішення знижують початкові інвестиції, але можуть вимагати більших витрат на підтримку. Хмарні сервіси мінімізують капітальні витрати, перетворюючи їх в операційні. Автоматизація зменшує потребу в людських ресурсах, що є значною статтею витрат для багатьох організацій.

Дотримання регуляторних вимог є обов'язковим для організацій в регульованих галузях. Законодавство про захист персональних даних вимагає забезпечення конфіденційності та цілісності інформації. Галузеві стандарти, такі як PCI DSS для платіжних систем, встановлюють специфічні вимоги до моніторингу безпеки. Можливість генерування звітів для аудиторів спрощує демонстрацію відповідності вимогам.

Адаптивність до еволюції загроз забезпечує довгострокову ефективність системи. Регулярні оновлення сигнатур та алгоритмів виявлення адаптують систему до нових типів атак. Можливість інтеграції нових джерел інформації про загрози збагачує контекст виявлення. Використання машинного навчання

дозволяє системі автоматично адаптуватися до змін в патернах мережевої активності.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз сучасних мережевих загроз та методів захисту інформаційних систем. Розглянуто основні типи мережевих атак, включаючи DDoS-атаки, атаки типу "людина посередині", SQL-ін'єкції та фішингові атаки. Проаналізовано існуючі системи виявлення та запобігання вторгненням (IDS/IPS), їх переваги та недоліки.

Встановлено, що сучасні системи захисту повинні поєднувати різні методи виявлення загроз, включаючи сигнатурний аналіз, виявлення аномалій та машинне навчання. Визначено основні вимоги до систем захисту від мережевих атак, зокрема щодо продуктивності, точності виявлення, масштабованості та зручності використання.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ СТАНУ МЕРЕЖІ ТА ВИЯВЛЕННЯ АТАК

2.1. Сценарії використання системи моніторингу

Проектування системи моніторингу мережевої безпеки розпочинається з детального аналізу сценаріїв використання, які визначають функціональні можливості та взаємодію системи з користувачами й іншими компонентами інфраструктури. Ідентифікація ключових акторів та їхніх ролей дозволяє сформулювати вимоги до інтерфейсів і розподілу повноважень.

Адміністратор безпеки є основним користувачем системи, відповідальним за конфігурацію, моніторинг та реагування на виявлені загрози. Типовий робочий день адміністратора починається з перегляду інформаційної панелі, що відображає поточний стан мережевої безпеки. Візуалізація включає графіки інтенсивності трафіку, розподіл типів виявлених атак, географічне походження загроз і статус критичних систем. Кольорове кодування дозволяє миттєво оцінити серйозність ситуації: зелений колір вказує на нормальну роботу, жовтий – на підозрілу активність, що потребує уваги, червоний – на активні атаки, які вимагають негайного втручання.

При виявленні підозрілої активності адміністратор переходить до детального аналізу конкретної події. Система надає комплексну інформацію: часові мітки початку та завершення активності, IP-адреси джерела й призначення з додатковою інформацією про географічне розташування та належність до організацій, використані порти й протоколи, характеристики трафіку. Контекстна інформація зі системи аналізу загроз збагачує розуміння небезпеки, показуючи, чи відомий IP-адрес атакуючого як джерело попередніх атак, чи асоційований він з відомими ботнетами або групами зловмисників, який тип загроз зазвичай походить із цього джерела [22-23].

Розслідування інциденту вимагає доступу до історичних даних для розуміння повного контексту атаки. Адміністратор використовує потужну систему пошуку для фільтрації подій за різними критеріями. Можливість побудови часової лінії активності дозволяє відстежити розвиток атаки від початкової розвідки до спроб експлуатації вразливостей. Кореляція подій із різних джерел виявляє зв'язки між, здавалося б, незалежними інцидентами, розкриваючи координовані багатовекторні атаки.

Реагування на підтверджену загрозу може здійснюватися через інтегровані механізми або вимагати ручного втручання. Для автоматизованої відповіді адміністратор налаштовує правила, що визначають дії при виявленні специфічних типів атак. Блокування IP-адрес джерела атаки на міжмережевому екрані або маршрутизаторі запобігає подальшій шкідливій активності. Ізоляція скомпрометованих вузлів від решти мережі обмежує потенційне поширення загрози. Автоматичне оповіщення відповідальних осіб забезпечує координацію реагування.

Налаштування системи виявлення є критичним процесом, що визначає баланс між чутливістю та кількістю хибних спрацьовувань. Адміністратор аналізує статистику виявлень для ідентифікації правил, які генерують надмірну кількість тривог низької важливості. Точне налаштування порогів виявлення аномалій враховує специфіку мережевої активності організації. Додавання винятків для легітимної активності, яка помилково класифікується як загроза, знижує рівень шуму. Створення власних правил виявлення дозволяє адаптувати систему до унікальних загроз, релевантних для конкретної організації.

Аналітик безпеки використовує систему для поглибленого дослідження тенденцій і патернів мережевої активності. Агреговані дані за тривалі періоди виявляють довгострокові тенденції: зміну географії джерел атак, еволюцію використовуваних методів, сезонні коливання інтенсивності загроз. Порівняльний аналіз між різними сегментами мережі ідентифікує компоненти,

що піддаються непропорційній кількості атак і можуть потребувати посиленого захисту.

Аналіз кореляцій між різними типами подій розкриває складні атаки, що складаються з численних етапів. Початкове сканування мережі для виявлення відкритих портів і сервісів може здаватися незначною подією, але якщо за ним сліднують спроби експлуатації вразливостей виявлених сервісів, це вказує на цілеспрямовану атаку. Виявлення таких ланцюжків дій дозволяє зупинити атаку на ранніх стадіях до того, як зловмисник досягне своїх цілей [24-25].

Генерування звітів є важливою функцією для комунікації з керівництвом і демонстрації відповідності регуляторним вимогам. Періодичні звіти про стан безпеки надають стислий огляд для керівництва, детальну статистику про виявлені загрози та вжиті заходи, оцінку ефективності систем захисту, рекомендації щодо покращення. Звіти про відповідність стандартам документують виконання специфічних контролів, що вимагаються регуляторами. Звіти про інциденти детально описують конкретні події безпеки для внутрішнього або зовнішнього аудиту.

Мережевий інженер взаємодіє з системою моніторингу для розуміння впливу мережевих змін на безпеку та діагностики проблем. Перед впровадженням нових сервісів або зміною мережевої топології інженер аналізує поточні патерни трафіку для прогнозування потенційного впливу. Після змін моніторинг дозволяє швидко виявити аномалії, що можуть вказувати на помилки конфігурації або непередбачені ефекти.

Діагностика мережевих проблем часто вимагає аналізу трафіку на рівні пакетів. Система надає інструменти для захоплення та аналізу детального трафіку між проблемними вузлами. Метрики продуктивності – затримки, втрати пакетів, повторні передачі – допомагають локалізувати джерело проблем. Історичні дані дозволяють порівнювати поточну поведінку з попереднім періодом коректної роботи для виявлення відхилень.

Автоматизована система реагування виконує попередньо визначені дії при виявленні специфічних загроз без втручання людини. Це критично важливо для атак, що розвиваються дуже швидко та вимагають негайної відповіді. Виявлення DDoS-атаки активує механізми пом'якшення наслідків, що перенаправляють трафік через системи очищення або активують захист на рівні провайдера. Ідентифікація зараженого вузла, що намагається поширити зловмисне програмне забезпечення, ініціює його автоматичну ізоляцію від мережі.

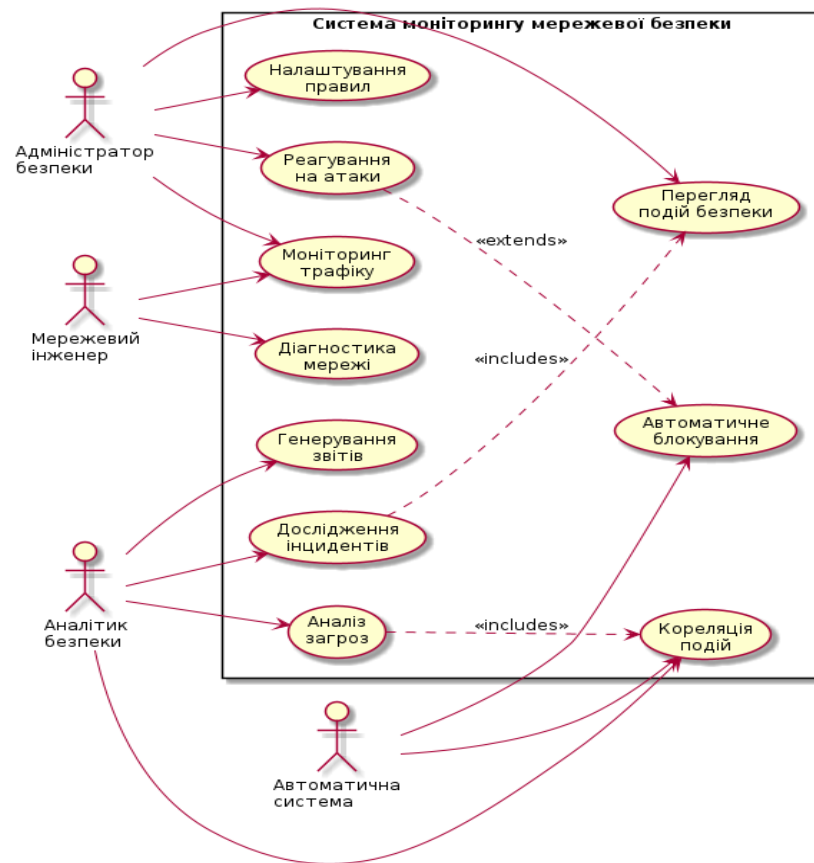


Рисунок 2.1 - Діаграма варіантів використання системи моніторингу

Інтеграція із системами оркестрації дозволяє координувати складні послідовності дій. Виявлення спроби несанкціонованого доступу може активувати робочий процес, що включає блокування облікового запису,

оповіщення адміністратора, збір додаткових доказів, створення запиту в системі управління інцидентами, оновлення бази загроз інформацією про атаку. Така автоматизація значно прискорює реагування та звільняє фахівців для роботи над складнішими завданнями.

Інтеграція з зовнішніми джерелами аналізу загроз збагачує контекст виявлених небезпек. Система автоматично запитує репутаційні бази даних для оцінки IP-адрес і доменів, що з'являються в трафіку. Виявлення комунікації з відомими командними серверами ботнетів або фішинговими сайтами негайно класифікується як високопріоритетна загроза. Регулярне оновлення індикаторів компрометації забезпечує захист від найновіших загроз.

Сценарій масштабної атаки демонструє поведінку системи під екстремальним навантаженням. Початок DDoS-атаки характеризується різким зростанням обсягу трафіку до цільового сервера. Система виявляє аномалію через порівняння з базовою лінією нормальної активності. Аналіз показує, що трафік походить із тисяч різних IP-адрес, багато з яких розташовані в географічних регіонах, нетипових для легітимних користувачів. Патерни запитів є однорідними, що вказує на їхнє автоматизоване генерування [26].

Автоматичне пом'якшення активує правила фільтрації, що відкидають трафік із характеристиками атаки. Обмеження швидкості лімітує кількість запитів з окремих джерел. Геоблокування відхиляє трафік із країн, де організація не має легітимних користувачів. Механізми перевірки, такі як JavaScript-валідація або капча, відфільтровують автоматизовані запити, зберігаючи доступ для реальних користувачів. Оповіщення ескалюється до провайдера для активації попереднього фільтрування, що знижує навантаження на канал зв'язку організації.

Сценарій складної цільової атаки ілюструє виявлення багатоступінчастої загрози. Початкове проникнення через цільовий фішинг може не викликати очевидних тривог. Однак аномальне виявлення фіксує незвичайну мережеву

активність із скомпрометованої робочої станції. Спроби бічного переміщення через мережу виявляються як нетипові з'єднання між системами, що зазвичай не взаємодіють. Ексфільтрація даних проявляється через аномальні обсяги вихідного трафіку до нових зовнішніх призначень.

Кореляція цих, здавалося б, незалежних подій виявляє зв'язок і дозволяє реконструювати повний ланцюжок атаки. Система автоматично створює граф атаки, що візуалізує розвиток загрози від початкової точки входу через внутрішнє поширення до спроб досягнення кінцевих цілей. Це дозволяє адміністраторам зрозуміти масштаб компрометації та вжити комплексних заходів для усунення загрози, включаючи ізоляцію всіх скомпрометованих систем, зміну облікових даних, перевірку на наявність прихованих входів.

2.2. Архітектура системи та структура модулів захисту

Архітектура системи моніторингу визначає організацію компонентів та їхню взаємодію для досягнення поставлених цілей виявлення та реагування на мережеві загрози. Проектування базується на принципах модульності для забезпечення гнучкості та масштабованості, розподіленості для покриття всіх сегментів мережі, надійності через резервування критичних компонентів.

Загальна архітектура системи організована за багатошаровою моделлю, де кожен рівень виконує специфічні функції та взаємодіє із суміжними рівнями через визначені інтерфейси. Шар збору даних відповідає за захоплення мережевого трафіку та подій із різних джерел. Шар обробки виконує аналіз зібраних даних, виявлення загроз, кореляцію подій. Шар зберігання забезпечує довготривале збереження даних для історичного аналізу та дотримання вимог про зберігання. Шар представлення надає інтерфейси для взаємодії з користувачами та зовнішніми системами.

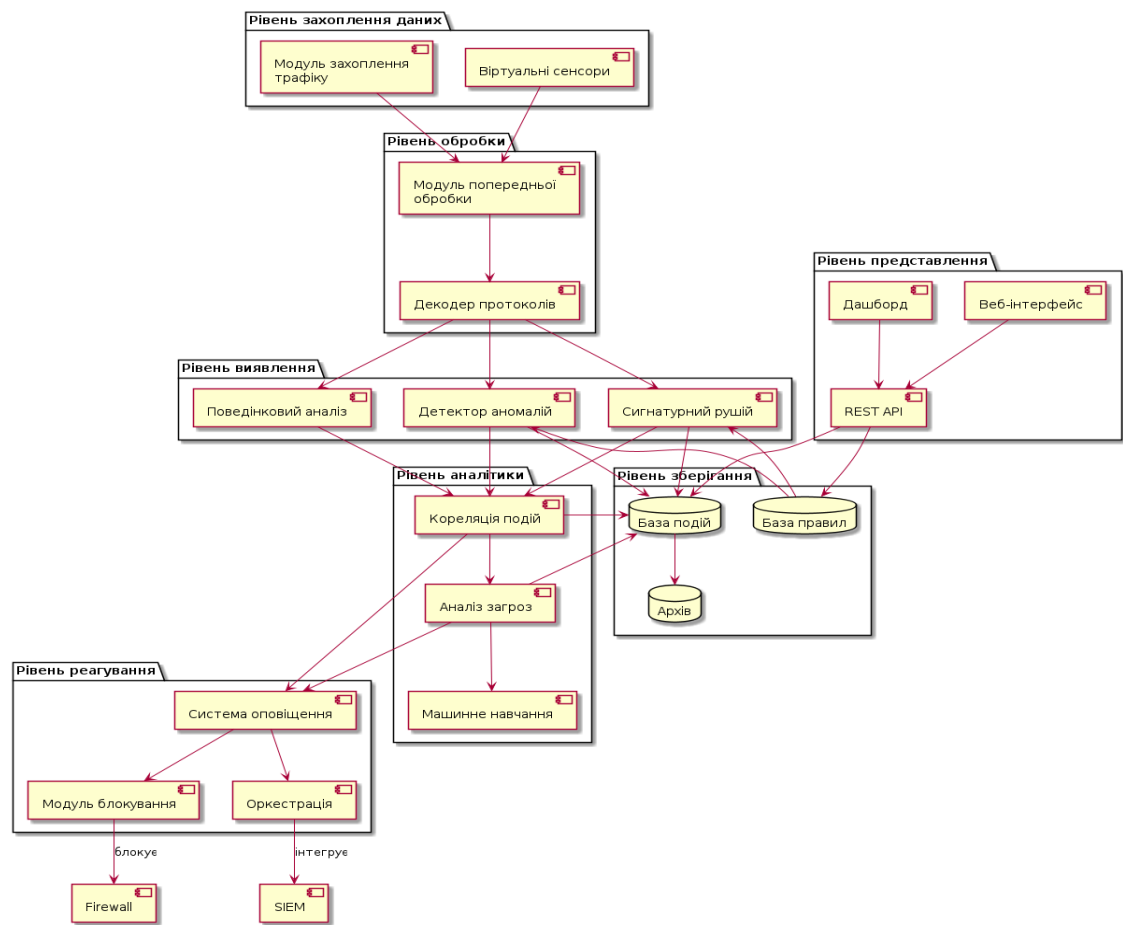


Рисунок 2.2 - Діаграма компонентів архітектури системи

Таблиця 2.1

Основні модулі системи та їх функції

Модуль	Функції
Захоплення трафіку	Отримання копії мережевого трафіку, інтеграція з мережевими TAP та SPAN
Попередня обробка	Дефрагментація IP, нормалізація протоколів, декодування застосунків
Виявлення загроз	Сигнатурне виявлення, детектор аномалій, поведінковий аналіз
Кореляція подій	Виявлення складних атак через патерни послідовностей подій

Зберігання даних	Індексування, архівування, управління життєвим циклом даних
Оповіщення	Інформування адміністраторів через email, SMS, webhook
Реагування	Автоматичне блокування, інтеграція з firewall, оркестрація workflow

Модуль захоплення трафіку є фундаментальним компонентом, що отримує копію мережевого трафіку для аналізу. Розміщення сенсорів у стратегічних точках мережі забезпечує покриття критичних сегментів. Периметральні сенсори моніторять трафік між внутрішньою мережею та Інтернетом, виявляючи спроби зовнішніх атак і несанкціоновану ексфільтрацію даних. Внутрішні сенсори розміщуються між сегментами мережі для виявлення бічного переміщення та внутрішніх загроз.

Технічна реалізація захоплення може використовувати різні підходи залежно від топології мережі та доступної інфраструктури. Дзеркальні порти на мережевих комутаторах копіюють трафік до порту моніторингу, де підключений сенсор. Це найпростіший метод, але він має обмеження пропускну здатності та може втрачати пакети при високому навантаженні. Мережеві відгалужувачі є спеціалізованими пристроями, що підключаються безпосередньо в розрив мережевого кабелю, забезпечуючи пасивне копіювання всього трафіку без ризику втрати пакетів або впливу на мережеву продуктивність [27].

Віртуальні сенсори інтегруються з гіпервізорами для моніторингу трафіку між віртуальними машинами. Це критично важливо в сучасних віртуалізованих середовищах, де значна частина комунікацій відбувається всередині фізичного сервера й не проходить через фізичні мережеві пристрої. Інтеграція з віртуальним комутатором дозволяє захоплювати весь трафік, включаючи внутрішню комунікацію між віртуальними машинами.

Модуль попередньої обробки нормалізує захоплений трафік перед передачею до аналізу. Дефрагментація IP-пакетів збирає фрагменти в повні пакети для коректного аналізу вмісту. Реасемблювання TCP-потоків об'єднує окремі сегменти в повну сесію, дозволяючи аналізувати прикладні протоколи незалежно від розбиття на пакети. Декодування протоколів витягує структуровану інформацію з бінарних даних, перетворюючи низькорівневі пакети в зрозумілі об'єкти прикладного рівня.

Нормалізація трафіку усуває неоднозначності, що можуть використовуватися зловмисниками для обходу систем виявлення. Різні операційні системи та застосунки по-різному інтерпретують певні аспекти протоколів, наприклад обробку невалідних полів заголовків або перекриваючихся фрагментів. Атакуючі можуть спеціально конструювати пакети, які будуть оброблені по-різному системою виявлення та цільовим вузлом, дозволяючи атаці пройти непоміченою. Нормалізація емулює поведінку цільових систем для забезпечення коректного виявлення.

Модуль виявлення загроз є серцем системи, що аналізує оброблений трафік для ідентифікації зловмисної активності. Сигнатурний рушій використовує базу правил, що описують відомі патерни атак. Кожне правило визначає умови, які повинні виконуватися для спрацювання: значення полів заголовків, наявність специфічних послідовностей байтів у навантаженні, характеристики потоку. Ефективна організація правил через індексування та оптимізацію забезпечує швидке співставлення навіть при великій кількості правил.

Детектор аномалій будує моделі нормальної поведінки мережі та виявляє відхилення, що можуть вказувати на загрози. Статистичний підхід обчислює базові метрики мережевої активності: обсяги трафіку між парами вузлів, розподіл використовуваних портів і протоколів, часові патерни активності. Значне відхилення від історичних середніх генерує тривогу для подальшого

розслідування [28-29]. Адаптивні пороги автоматично коригуються відповідно до довгострокових тенденцій, враховуючи легітимні зміни в патернах використання мережі.

Поведінковий аналіз моделює типові дії користувачів і застосунків. Машинне навчання на історичних даних створює профілі нормальної поведінки для різних типів активності. Виявлення дій, що не відповідають профілю, може вказувати на компрометацію облікового запису або системи. Наприклад, користувач, який зазвичай працює тільки з офісними документами та раптом починає виконувати адміністративні команди або звертатися до сервера баз даних безпосередньо, демонструє підозрілу поведінку.

Аналіз протоколів перевіряє відповідність трафіку специфікаціям протоколів і виявляє аномалії, що можуть вказувати на атаки або неправильні конфігурації. Валідація полів заголовків ідентифікує пакети з невалідними або суперечливими значеннями. Аналіз послідовностей перевіряє, що обмін даними слідує очікуваному автомату станів протоколу. Виявлення несподіваних команд або відповідей може вказувати на спроби експлуатації вразливостей реалізацій протоколів.

Модуль кореляції подій об'єднує інформацію з різних джерел для виявлення складних атак, що складаються з численних етапів. Часова кореляція ідентифікує події, що відбуваються в близькі моменти часу і можуть бути пов'язані. Просторова кореляція виявляє події, що стосуються одних і тих же вузлів або користувачів. Причинна кореляція встановлює логічні зв'язки між подіями, визначаючи, чи може одна подія бути наслідком іншої.

Правила кореляції описують патерни складних атак через послідовності або комбінації простіших подій. Наприклад, правило може визначати складну цільову атаку як послідовність: спочатку фішинг, потім встановлення прихованого входу, наступний збір облікових даних, бічне переміщення до критичних систем і, нарешті, ексфільтрація даних. Виявлення такої

послідовності, навіть якщо окремі події здаються незначними, сигналізує про серйозну загрозу [30].

Модуль аналізу загроз інтегрує інформацію із зовнішніх джерел для збагачення контексту виявлених загроз. Репутаційні бази даних надають оцінки IP-адрес, доменів і хешів файлів на основі попередньої зловмисної активності. Індикатори компрометації з каналів аналізу загроз дозволяють швидко виявляти відомі небезпеки. Інтеграція зі стандартами обміну інформацією забезпечує стандартизований обмін інформацією про загрози з іншими організаціями та платформами.

Локальна база аналізу загроз акумулює знання про небезпеки, специфічні для організації. Історія атак та їхні характеристики дозволяють виявляти повторювані патерни. Список внутрішніх індикаторів компрометації включає відомі зловмисні IP-адреси, що атакували організацію, хеші зловмисного програмного забезпечення, виявленого в мережі, домени фішингових сайтів, націлених на співробітників. Регулярне оновлення та очищення бази забезпечує її актуальність.

Модуль зберігання даних управляє збереженням великих обсягів інформації про мережеву активність і події безпеки. Багаторівнева архітектура оптимізує баланс між швидкістю доступу та вартістю зберігання. Гарячий рівень зберігає дані за останні дні або тижні на швидких твердотільних накопичувачах для забезпечення оперативного доступу при розслідуваннях і реагуванні. Теплий рівень містить дані за останні місяці на традиційних жорстких дисках для історичного аналізу та генерування звітів. Холодний рівень архівує старі дані на дешевому сховищі або в хмарі для довгострокового зберігання відповідно до регуляторних вимог.

Індексування даних забезпечує швидкий пошук за різними критеріями. Повнотекстовий пошук дозволяє знаходити події, що містять специфічні рядки або патерни. Індeksi за часовими мітками оптимізують вибірки за часовими

діапазонами. Індокси за IP-адресами, користувачами, типами подій прискорюють типові запити. Стиснення зменшує обсяги зберігання, зберігаючи можливість ефективного пошуку [31].

Модуль аналітики надає інструменти для інтерактивного дослідження даних і виявлення патернів. Мова запитів дозволяє формувати складні умови вибірки та агрегації даних. Візуалізації включають часові графіки для аналізу тенденцій, географічні карти для відображення джерел загроз, мережеві графі для розуміння комунікаційних патернів, теплові карти для виявлення аномалій у розподілах. Збережені пошуки та інформаційні панелі дозволяють швидко повторювати типові аналізи.

Статистичні функції обчислюють агреговані метрики за довільними часовими вікнами та групуваннями. Підрахунок кількості подій різних типів показує розподіл загроз. Обчислення сум і середніх для метрик на кшталт обсягів трафіку виявляє аномалії. Виявлення топ найактивніших джерел або цілей атак ідентифікує найбільш атаковані або атакуючі системи. Порівняння між часовими періодами виявляє зміни в патернах загроз.

Модуль оповіщення забезпечує своєчасне інформування про критичні події безпеки. Правила оповіщення визначають умови, що запускають нотифікації: виявлення атак певних типів, перевищення порогів інтенсивності або аномалій, зміна стану критичних систем. Канали доставки включають електронну пошту для неургентних тривог, текстові повідомлення для високопріоритетних подій, що вимагають негайної уваги, інтеграцію з системами управління інцидентами для автоматичного створення запитів, вебхуки для інтеграції з іншими системами автоматизації.

Інтелектуальна фільтрація оповіщень запобігає перевантаженню адміністраторів. Дедуплікація об'єднує ідентичні або дуже схожі тривоги, що виникають у короткий період, в одне оповіщення. Агрегація групує пов'язані події, наприклад численні спроби доступу з одного джерела, в сумарне

оповіщення. Придушення тимчасово вимикає оповіщення про відомі проблеми, що вже розслідуються або для яких заплановане обслуговування. Ескалація підвищує пріоритет оповіщень, що ігноруються тривалий час.

Модуль реагування виконує автоматизовані дії у відповідь на виявлені загрози. Блокування на рівні мережі додає правила на міжмережеві екрани або маршрутизатори для відхилення трафіку зі зловмисних джерел. Обмеження швидкості лімітує кількість запитів, що приймаються від підозрілих джерел, без повного блокування. Ізоляція скомпрометованих вузлів переміщує їх у карантинну віртуальну локальну мережу з обмеженим доступом для запобігання поширенню загрози. Завершення підозрілих процесів або з'єднань на захищених вузлах може зупинити атаку в процесі розвитку.

Інтеграція з платформами оркестрації координує складні робочі процеси реагування, що включають численні системи та ручні кроки. Сценарій визначає послідовність дій для реагування на специфічні типи інцидентів. Автоматизація рутинних задач – збір додаткової інформації, оновлення конфігурацій, документування дій – прискорює реагування та забезпечує консистентність. Ручні затвердження для критичних дій балансують автоматизацію з людським контролем.

Модуль звітності генерує різноманітні звіти для різних аудиторій. Оперативні звіти надають детальну технічну інформацію для адміністраторів безпеки: списки всіх подій, деталі виявлених атак, вжиті дії. Звіти для керівництва представляють огляд високого рівня стану безпеки в нетехнічній формі, з фокусом на бізнес-впливах і тенденціях. Звіти про відповідність документують виконання специфічних контролів для демонстрації відповідності регуляторним вимогам.

Шаблони звітів визначають структуру та зміст для різних типів звітів. Параметризація дозволяє налаштовувати часові діапазони, фільтри, рівень деталізації. Періодична автоматична генерація забезпечує регулярну доставку

звітів без ручного втручання. Експорт у різні формати – PDF для розповсюдження, CSV для подальшого аналізу в інших інструментах, HTML для перегляду в браузері – задовольняє різні потреби користувачів.

Модуль адміністрування надає інструменти для управління самою системою моніторингу. Управління користувачами включає створення та деактивацію облікових записів, призначення ролей і дозволів, налаштування автентифікації, включно з інтеграцією з корпоративними службами каталогів. Аудит дій адміністраторів забезпечує підзвітність через журналювання всіх змін конфігурації та доступу до чутливих даних.

Управління конфігурацією включає налаштування параметрів роботи всіх модулів системи. Версіонування конфігурацій дозволяє відслідковувати зміни та повертатися до попередніх станів при проблемах. Резервне копіювання регулярно зберігає конфігурації та критичні дані для відновлення після збоїв. Моніторинг стану системи відслідковує статус всіх компонентів, використання ресурсів, продуктивність, генеруючи тривоги при виявленні проблем.

2.3. Алгоритми виявлення мережевих аномалій та атак

Ефективне виявлення мережевих загроз базується на комбінації різноманітних алгоритмів, що доповнюють один одного, компенсуючи слабкі сторони окремих підходів. Сигнатурні методи забезпечують високу точність виявлення відомих загроз, аномальні алгоритми виявляють невідомі атаки, поведінкові підходи розпізнають складні багатоетапні загрози.

Алгоритм сигнатурного виявлення базується на співставленні мережевого трафіку з базою відомих патернів атак. Кожна сигнатура описує характеристики специфічної атаки через набір умов, які повинні виконуватися для детекції. Умови можуть стосуватися значень полів заголовків протоколів, наявності специфічних байтових послідовностей у навантаженні пакетів, характеристик потоку, таких як напрямок передачі даних або розміри пакетів.

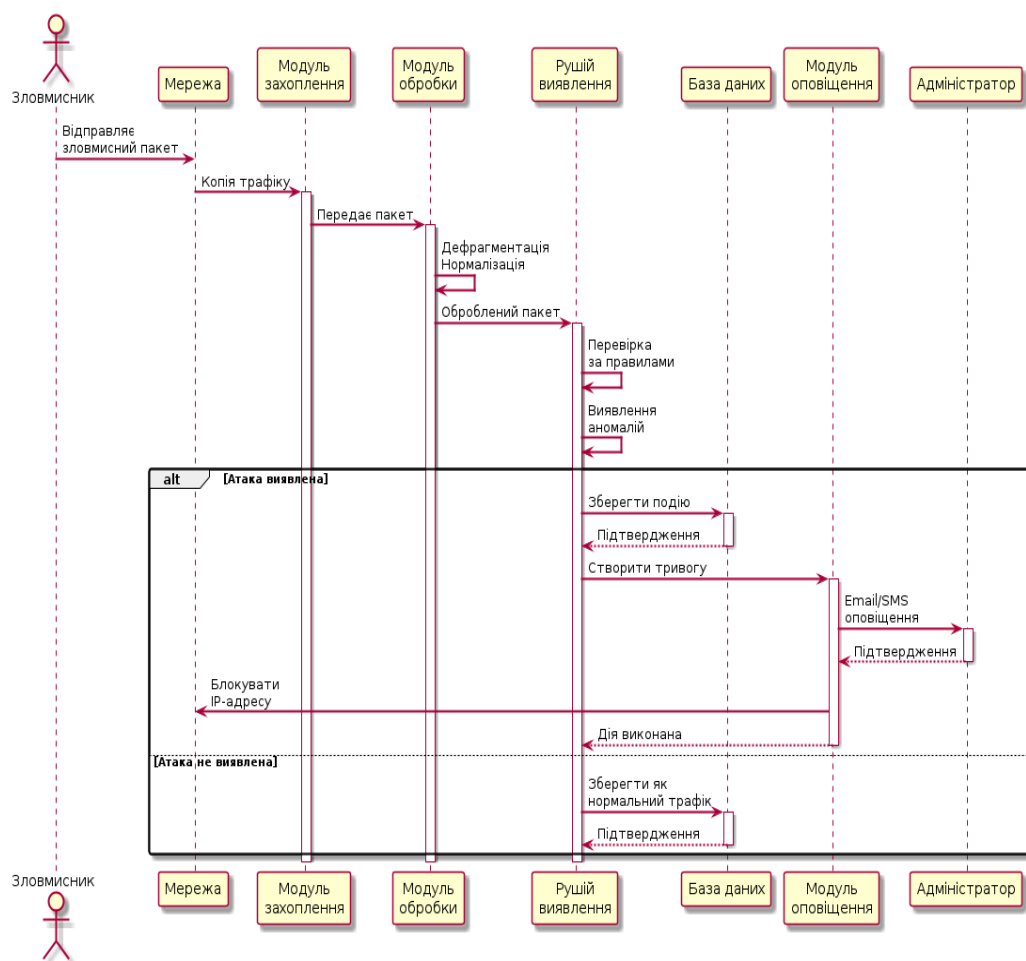


Рисунок 2.3 - Діаграма послідовності процесу виявлення атаки

Структура правила виявлення включає заголовок, що визначає дію при спрацюванні, протокол, адреси та порти джерела й призначення. Опції правила специфікують детальніші умови співставлення. Контентні опції перевіряють наявність або відсутність специфічних рядків або байтових послідовностей у корисному навантаженні. Модифікатори контенту визначають місце пошуку, чутливість до регістру, інтерпретацію як шістнадцяткові дані або текст. Неконтентні опції перевіряють метадані пакетів: розміри, прапорці TCP-заголовків, час життя пакета.

Оптимізація співставлення з великою кількістю правил є критично важливою для забезпечення продуктивності. Індекссування правил організовує їх у структури даних, що дозволяють швидко відкидати неприйнятні правила.

Групування за протоколами та портами дозволяє перевіряти тільки релевантні підмножини правил для кожного пакета. Алгоритми швидкого пошуку підрядків ефективно знаходять байтові послідовності в навантаженні навіть при великій кількості патернів.

Алгоритм виявлення аномалій базової лінії будує статистичну модель нормальної мережевої активності та виявляє відхилення, що перевищують визначені пороги. Період навчання агрегує метрики трафіку за історичний період, що представляє нормальну роботу без атак. Для кожної метрики обчислюються статистичні параметри: середнє значення, стандартне відхилення, мінімум і максимум, квантилі розподілу [32-33].

Моніторині метрики включають обсяги трафіку як загальні, так і розподілені за протоколами, кількість пакетів і з'єднань, розподіли розмірів пакетів і тривалості з'єднань, співвідношення вхідного та вихідного трафіку, інтенсивність використання різних портів і сервісів. Метрики обчислюються на різних рівнях агрегації: глобально для всієї мережі, для окремих підмереж або віртуальних локальних мереж, для індивідуальних вузлів, для пар комунікуючих вузлів.

Виявлення аномалій порівнює поточні значення метрик із моделлю базової лінії. Простий пороговий метод генерує тривогу, коли метрика виходить за межі діапазону нормальних значень, наприклад перевищує середнє плюс кілька стандартних відхилень. Складніші підходи враховують часову динаміку, перевіряючи, чи відхилення є короткочасним сплеском або стійкою тенденцією. Адаптивні пороги автоматично коригуються з часом для врахування довгострокових змін у патернах використання мережі.

Алгоритм виявлення DDoS-атак спеціалізується на розпізнаванні різних типів атак на відмову в обслуговуванні. Об'ємні атаки виявляються через моніторинг загальних обсягів трафіку та виявлення різких зростань, що значно перевищують нормальні рівні. Аналіз розподілу джерел трафіку виявляє, чи

атака походить із численних різних IP-адрес, що характерно для розподілених атак. Географічний аналіз перевіряє, чи джерела атаки розташовані в регіонах, нетипових для легітимних користувачів організації.

Протокольні атаки виявляються через аномалії у використанні протоколів. Атака переповненням запитами на встановлення з'єднання детектується через моніторинг балансу між пакетами ініціалізації з'єднання та пакетами завершення. Значне перевищення ініціювань без відповідних завершень вказує на атаку. Збільшення кількості напіввідкритих з'єднань на серверах також є індикатором. Фрагментаційні атаки виявляються через аномальні розподіли розмірів фрагментів, надмірну кількість фрагментованих пакетів, підозрілі перекриття фрагментів.

Атаки на рівні застосунків є найскладнішими для виявлення, оскільки імітують легітимний трафік. Аналіз патернів запитів виявляє аномалії: дуже однорідні рядки ідентифікації користувацького агента, що вказують на автоматизацію, відсутність типових заголовків, присутніх у запитах реальних браузерів, аномальні послідовності запитів, що не відповідають типовій поведінці користувачів. Аналіз сесій виявляє короткоживучі з'єднання з великою кількістю запитів, характерні для ботів [34].

Алгоритм виявлення сканування портів ідентифікує спроби розвідки мережі, що часто передують цільовим атакам. Вертикальне сканування характеризується спробами з'єднання з численними різними портами на одному вузлі в короткий період. Горизонтальне сканування намагається з'єднатися з одним і тим же портом на численних різних вузлах. Моніторинг співвідношення невдалих спроб з'єднання до успішних виявляє сканування, оскільки більшість портів зазвичай закриті.

Приховані техніки сканування намагаються уникнути виявлення через уповільнення швидкості або використання специфічних типів пакетів. Синхронне сканування не завершує рукописання протоколу, надсилаючи

тільки пакети ініціалізації та аналізуючи відповіді. Сканування з використанням нестандартних комбінацій прапорців використовують пакети з незвичайними налаштуваннями. Виявлення базується на розпізнаванні незвичайних патернів комунікації, що не відповідають нормальним сесіям.

Розподілене сканування координує атаку з численних джерел для уникнення виявлення за кількістю спроб з одного адресу. Кореляція спроб із різних джерел, спрямованих на один і той же набір цілей, виявляє таку координацію. Часова кореляція ідентифікує сканування, що виконується дуже повільно, розтягуючи спроби на години або дні для уникнення порогових детекторів.

Алгоритм виявлення SQL-ін'єкцій аналізує HTTP-запити на наявність SQL-синтаксису в параметрах. Регулярні вирази розпізнають типові SQL-конструкції: об'єднання вибірок, завжди істинні умови, символи коментарів, що використовуються для обходу валідації. Аналіз контексту перевіряє, чи очікується наявність таких рядків у легітимних запитах до даного застосунку. Виявлення численних спроб із різними ін'єкціями вказує на автоматизоване тестування.

Сліпі SQL-ін'єкції є складнішими для виявлення, оскільки не викликають очевидних помилок. Аналіз часових затримок відповідей виявляє часові сліпі ін'єкції, де атакуючий використовує команди затримки для витягування інформації по одному біту. Порівняння розмірів відповідей виявляє логічні сліпі ін'єкції, де інформація витягується через спостереження за змінами в контенті відповідей [35-36].

Алгоритм виявлення міжсайтового скриптингу шукає спроби впровадження JavaScript-коду в параметри, що будуть відображені на веб-сторінках. Патерни включають теги скриптів, обробники подій на кшталт завантаження або помилки, псевдопротокол JavaScript у посиланнях.

Обфускація через URL-кодування, HTML-сутності, різні методи екранування потребує нормалізації перед аналізом для виявлення замаскованих атак.

Алгоритм виявлення зловмисного програмного забезпечення в трафіку аналізує комунікаційні патерни заражених вузлів. Трафік командування та контролю часто має характеристики: періодичність сигналів маяка з регулярними інтервалами, комунікація з підозрілими доменами, що є молодими або розташовані в екзотичних доменах верхнього рівня, використання нестандартних портів для відомих протоколів, шифрування трафіку застосунків, що зазвичай працюють у відкритому тексті.

Ексфільтрація даних виявляється через аномально великі обсяги вихідного трафіку, особливо в нетиповий час або до нових призначень. Аналіз ентропії навантаження виявляє зашифровані або упаковані дані, що можуть бути викраденою інформацією. Порівняння з типовими патернами передачі файлів ідентифікує аномалії в характері даних.

Алгоритм поведінкового профілювання користувачів створює моделі типової активності для виявлення скомпрометованих облікових записів. Профіль включає типові часи роботи, використовувані системи та сервіси, обсяги й напрямки мережевої активності, географічне розташування доступу. Значні відхилення від профілю – доступ у нетиповий час, з нової географічної локації, до систем, до яких користувач раніше не звертався – генерують тривоги [37-38].

Таблиця 2.2

Метрики базової лінії для виявлення аномалій

Метрика	Параметр	Поріг відхилення
Обсяг трафіку	Байти/секунду	$\pm 3\sigma$ від середнього
Кількість з'єднань	З'єднань/хвилину	$\pm 3\sigma$ від середнього
Розподіл портів	Ентропія розподілу	>20% від базової
Частота помилок	Помилки/хвилину	>50% від базової

Розмір пакетів	Середній розмір (байти)	$\pm 3\sigma$ від середнього
----------------	-------------------------	------------------------------

Аномальні дії включають підвищення привілеїв без бізнес-обґрунтування, доступ до великої кількості записів у базах даних, завантаження чутливих документів, використання адміністративних інструментів звичайними користувачами. Машинне навчання на історичних даних про дії користувачів автоматично будує складні моделі, що враховують численні фактори та їхні взаємодії.

2.4. Модель даних та структура інформаційного забезпечення

Ефективна система моніторингу потребує добре спроектованої моделі даних, що забезпечує ефективне зберігання, швидкий пошук і аналіз величезних обсягів інформації про мережеву активність та події безпеки. Модель даних визначає структуру інформації, зв'язки між різними типами об'єктів, індекси для оптимізації запитів.

Концептуальна модель організує інформацію навколо основних сутностей домену мережевої безпеки. Подія безпеки є центральним об'єктом, що представляє будь-яку значущу активність, виявлену системою. Атрибути події включають часову мітку з мілісекундною точністю, тип події з ієрархічної класифікації, важливість або рівень критичності, статус обробки. Кожна подія пов'язана з іншими сутностями, що надають контекст.

Мережева комунікація описує обмін даними між двома кінцевими точками. Атрибути включають IP-адреси джерела та призначення з додатковою інформацією про їхню резолюцію в доменні імена, географічну локацію, приналежність до організацій. Порти та протоколи ідентифікують сервіси, що комунікують. Метрики комунікації включають кількість пакетів і байтів у кожному напрямку, тривалість з'єднання, часові характеристики. Дані

навантаження можуть зберігатися для детального аналізу або реконструкції сесій.

Вузол представляє мережевий пристрій в інфраструктурі. Атрибути включають численні IP-адреси, якщо пристрій має кілька інтерфейсів, фізичні адреси для ідентифікації на каналному рівні, доменне ім'я, тип пристрою та операційна система, визначені через цифровий відбиток, роль в інфраструктурі, власник або відповідальна особа. Історія виявлених сервісів і відкритих портів допомагає розуміти функції вузла.

Користувач представляє особу, що працює з системами. Атрибути включають ідентифікатор користувача, повне ім'я та контактні дані, роль і підрозділ в організації, призначені системи та дозволи. Зв'язок між користувачами та вузлами відслідковує, де та коли користувач працював. Профіль поведінки зберігає типові патерни активності для аномального виявлення.

Загроза або тип атаки описує класи зловмисної активності. Атрибути включають ідентифікатор у таксономіях, опис характеристик і цілей загрози, тактики, техніки та процедури, використовувані атакуючими, індикатори компрометації, характерні для цієї загрози. Зв'язки з вразливостями показують, які слабкості експлуатує загроза. Контрзаходи описують методи детекції та пом'якшення наслідків [39].

Вразливість представляє відому слабкість у програмному або апаратному забезпеченні. Атрибути включають стандартний ідентифікатор вразливості, опис природи вразливості, оцінку серйозності, вплинуті версії програмного забезпечення, доступні виправлення або тимчасові рішення. Зв'язок із вузлами вказує, які системи в інфраструктурі є вразливими.

Актор представляє джерело загроз — від окремих зловмисників до організованих груп. Атрибути включають ідентифікатори в базах аналізу загроз, відомі мотивації та цілі, типові тактики та інструменти, географічна або

галузева спрямованість. Зв'язок із загрозами показує, які типи атак асоціюються з актором.

Артефакт представляє технічний індикатор загрози. Типи включають IP-адреси відомих зловмисних вузлів, домени командних серверів, посилання фішингових або сайтів розповсюдження зловмисного програмного забезпечення, файлові хеші зловмисного програмного забезпечення, сертифікати шифрування, адреси електронної пошти. Атрибути включають тип артефакту, значення, рівень довіри до індикатора, джерело інформації, період валідності.

Інцидент безпеки агрегує пов'язані події в єдиний випадок, що вимагає розслідування та реагування. Атрибути включають унікальний ідентифікатор, тип інциденту, поточний статус, пріоритет, часові мітки створення та закриття, призначений відповідач. Зв'язки з подіями показують всі виявлення, що є частиною інциденту. Часова лінія документує хронологію дій у процесі розслідування та реагування [40].

Правило виявлення описує логіку ідентифікації загроз. Атрибути включають унікальний ідентифікатор, тип правила, умови спрацювання, дії при виявленні, метаінформацію про автора та дату створення. Зв'язок із загрозами показує, які атаки призначене виявляти правило. Статистика спрацьовувань допомагає оцінювати ефективність і налаштовувати пороги.

Фізична модель даних реалізує концептуальну модель у конкретній системі зберігання з урахуванням вимог до продуктивності та масштабованості. Гібридний підхід поєднує різні типи баз даних для оптимального зберігання різних видів інформації. Часові ряди зберігають метрики та події, що є природно впорядкованими за часом. Спеціалізовані бази даних часових рядів оптимізовані для швидкого запису та агрегацій за часовими вікнами.

Документне сховище зберігає події безпеки як документи з гнучкою схемою, що дозволяє легко додавати нові поля без зміни структури. Пошукова система забезпечує потужний повнотекстовий пошук і агрегації для аналізу.

Індексування всіх полів дозволяє швидко фільтрувати за будь-якими критеріями. Розподіл даних між численними вузлами забезпечує масштабування.

Реляційна база даних зберігає структуровані довідкові дані: конфігурації, користувачі, правила, зв'язки між сутностями. Нормалізована схема забезпечує консистентність і ефективність зберігання. Транзакційна семантика гарантує цілісність при конкурентних змінах. Складні запити з об'єднанням дозволяють витягувати зв'язану інформацію [41].

Графова база даних моделює зв'язки між сутностями як граф для ефективного аналізу складних взаємозв'язків. Вузли представляють хости, користувачів, події, загрози. Ребра представляють комунікації між вузлами, дії користувачів, зв'язки між подіями в атаках. Графові запити ефективно знаходять патерни та шляхи в мережі взаємозв'язків.

Індексування є критично важливим для забезпечення швидкого пошуку у великих обсягах даних. Первинний індекс за часовими мітками є фундаментальним, оскільки більшість запитів обмежуються часовими діапазонами. Розподіл даних за часом дозволяє ефективно видаляти старі дані та оптимізує запити, що покривають обмежені періоди. Вторинні індекси за часто використовуваними полями прискорюють фільтрацію.

Композитні індекси за комбінаціями полів оптимізують типові запити, що фільтрують за численними критеріями одночасно. Наприклад, індекс за парою IP-адрес джерела та призначення прискорює пошук комунікацій між специфічними вузлами. Індекс за типом події та критичністю дозволяє швидко знаходити високопріоритетні загрози певних типів.

Повнотекстові індекси дозволяють шукати текстові рядки в полях, що містять довільний текст: описи подій або дані навантаження. Аналіз тексту розбиває рядки на токени, нормалізує їх, будує інвертований індекс, що зв'язує терміни з документами, де вони зустрічаються. Це дозволяє швидко знаходити документи, що містять специфічні слова або фрази.

Архівування та життєвий цикл даних управляє зберіганням протягом часу з балансуванням між доступністю та вартістю. Гарячий рівень зберігає дані за останні дні або тижні на швидких твердотільних накопичувачах для негайного доступу при розслідуваннях. Теплий рівень переміщує старіші дані на звичайні диски, де вони все ще доступні, але з дещо нижчою продуктивністю. Холодний рівень архівує дуже старі дані на дешеві носії або в хмарне сховище з можливістю відновлення при потребі.

Ротація індексів створює нові індекси періодично, наприклад щоденно, що дозволяє ефективно видаляти старі дані, просто видаляючи цілі індекси замість окремих записів. Це також оптимізує продуктивність, оскільки індекси залишаються розумного розміру. Автоматичне перемикання створює новий індекс, коли поточний досягає певного розміру або віку [42].

Стиснення зменшує обсяги зберігання, особливо для архівних даних, що рідко використовуються. Колонкове стиснення використовує той факт, що дані в одному полі часто мають схожі значення, що добре стискаються. Словникове кодування замінює повторювані значення посиланнями на словник. Для дуже холодного сховища можна використовувати агресивніше стиснення, що вимагає більше обчислень для розпакування, але економить багато місця.

2.5. Об'єктна структура програмного комплексу

Проектування програмного комплексу системи моніторингу базується на об'єктно-орієнтованому підході, що забезпечує модульність, розширюваність і підтримуваність коду. Архітектура організована навколо чітко визначених класів та інтерфейсів, що інкапсулюють функціональність і взаємодіють через визначені контракти.

Пакетна структура організовує класи в логічні групи відповідно до їхньої функціональності. Пакет захоплення містить класи, відповідальні за захоплення мережевого трафіку з різних джерел. Пакет обробки включає компоненти

попередньої обробки та нормалізації даних. Пакет виявлення містить рушії виявлення загроз. Пакет зберігання забезпечує персистентність даних. Пакет аналітики надає функціональність аналізу. Пакет представлення включає веб-інтерфейс і програмний інтерфейс. Пакет інтеграції забезпечує взаємодію з зовнішніми системами.

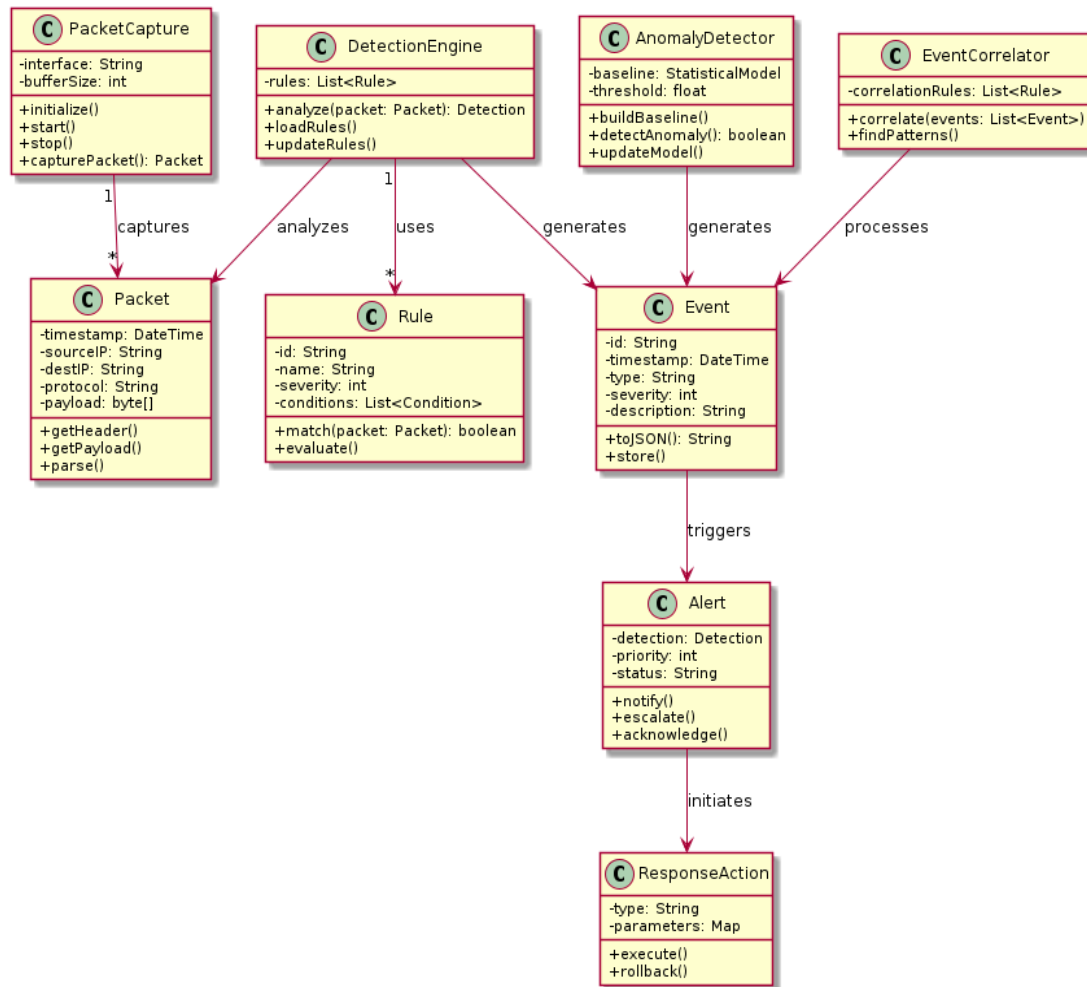


Рисунок 2.4 - Діаграма класів системи виявлення загроз

Базовий клас захоплення мережі визначає інтерфейс для захоплення трафіку. Абстрактні методи: ініціалізація конфігурує джерело захоплення, запуск розпочинає моніторинг, зупинка припиняє захоплення, отримання пакетів повертає захоплені пакети. Конкретні реалізації використовують різні технології

захоплення з мережевого інтерфейсу, отримують дані з експортерів потоків, працюють із дзеркальними портами комутаторів.

Клас пакета інкапсулює мережевий пакет і надає методи для доступу до його компонентів. Атрибути включають часову мітку захоплення, бінарний вміст пакета, розібрані заголовки різних рівнів, дані прикладного рівня. Методи отримання адрес витягують IP-адреси джерела та призначення, метод отримання протоколу повертає протокол, метод отримання навантаження надає доступ до даних. Відкладений розбір затримує розбір заголовків до першого звернення для оптимізації продуктивності.

Клас обробника пакетів виконує попередню обробку та нормалізацію пакетів. Метод дефрагментації збирає фрагментовані пакети в повні датаграми. Метод реасемблювання потоку об'єднує сегменти в повний потік даних застосунку. Метод нормалізації протоколу усуває неоднозначності в інтерпретації протоколів для забезпечення коректного виявлення. Шаблон конвеєра дозволяє конфігурувати послідовність обробників, що застосовуються до кожного пакета.

Базовий інтерфейс рушія виявлення визначає контракт для рушіїв виявлення. Метод аналізу приймає оброблені мережеві дані та повертає список виявлених загроз. Метод оновлення правил оновлює набір правил виявлення. Різні реалізації: сигнатурний рушій використовує сигнатурне співставлення, рушій аномалій застосовує аномальне виявлення, поведінковий рушій аналізує поведінкові патерни.

Клас правила представляє правило виявлення. Атрибути включають унікальний ідентифікатор, людино-читабельну назву, опис загрози, важливість, умови спрацювання, дії при виявленні. Метод співставлення перевіряє, чи задовольняє подія умовам правила. Фабричний метод створення з рядка розбирає текстове представлення правила в об'єкт.

Клас умови представляє умову в правилі виявлення. Різні підкласи реалізують специфічні типи умов: умова IP-адреси перевіряє адреси, умова порту перевіряє порти, умова вмісту шукає байтові послідовності в навантаженні, умова порогу перевіряє перевищення порогів. Композитні умови комбінують прості умови в складну логіку через логічні операції.

Клас виявлення представляє виявлену загрозу. Атрибути включають часову мітку виявлення, правило, що спрацювало, важливість, джерело атаки, ціль атаки, докази, що підтверджують виявлення. Методи: ескалація підвищує пріоритет, призначення визначає відповідача, розв'язання позначає як оброблене.

Клас детектора аномалій реалізує виявлення аномалій. Метод побудови базової лінії будує статистичну модель нормальної активності на історичних даних. Метод виявлення аномалії порівнює поточні метрики з базовою лінією та повертає оцінку аномальності. Атрибути базової лінії зберігають статистичні параметри, пороги визначають межі для різних рівнів критичності. Адаптивне навчання періодично оновлює базову лінію для врахування змін у легітимній активності.

Клас збирача метрик збирає агреговані метрики мережевої активності. Методи: збільшення лічильника інкрементує лічильник, запис значення фіксує значення метрики, отримання метрик повертає зібрані дані. Виконання підтримує агрегацію за часовими вікнами різної тривалості. Структури в пам'яті забезпечують швидкий доступ, періодичне скидання зберігає дані в постійне сховище.

Інтерфейс сховища даних визначає абстракцію для зберігання даних. Методи: збереження зберігає запис, отримання повертає за ідентифікатором, запит виконує складний пошук, видалення видаляє дані. Різні реалізації для часових метрик, для подій, для структурованих даних дозволяють використовувати оптимальні технології для різних типів інформації.

Клас події представляє подію безпеки в системі. Атрибути включають унікальний ідентифікатор, часову мітку, тип події, важливість, джерело, призначення, додаткову інформацію. Методи: серіалізація в формат обміну даними для програмного інтерфейсу, десеріалізація з формату, збагачення додає додатковий контекст із аналізу загроз.

Клас корелятора подій виконує кореляцію подій для виявлення складних атак. Метод кореляції приймає нову подію та шукає зв'язки з попередніми подіями. Правила кореляції визначають патерни, що складаються з послідовностей або комбінацій подій. Часові вікна обмежують період, у якому події вважаються пов'язаними. Відслідковування стану підтримує стан для виявлення багатоетапних атак.

Клас аналізу загроз інтегрує зовнішні джерела інформації про загрози. Методи збагачення запитують репутацію IP-адреси, перевіряють домен, шукають інформацію про файловий хеш. Кешування зменшує кількість зовнішніх запитів і прискорює збагачення. Різні провайдери можуть використовуватися через єдиний інтерфейс.

Клас тривоги представляє оповіщення про подію безпеки. Атрибути включають виявлення, що викликало тривогу, одержувачі, канал доставки, статус обробки. Методи: відправлення надсилає оповіщення, підтвердження фіксує отримання, ескалація підвищує при ігноруванні. Шаблони форматують повідомлення для різних каналів.

Клас менеджера тривог управляє життєвим циклом оповіщень. Метод створення тривоги формує нове оповіщення з виявлення, дедуплікація об'єднує ідентичні тривоги, обмеження швидкості лімітує частоту для запобігання перевантаженню. Правила маршрутизації визначають, хто та коли повинен отримувати оповіщення різних типів. Інтеграції забезпечують доставку через різні канали.

Клас дії реагування представляє дію у відповідь на загрозу. Різні підкласи реалізують специфічні дії: блокування адреси додає правило блокування, ізоляція вузла переміщує вузол у карантин, завершення процесу припиняє процес, збір доказів збирає додаткові докази. Атрибути: ціль ідентифікує об'єкт дії, параметри конфігурують виконання, статус відслідковує результат.

Клас рушія сценаріїв автоматизує виконання складних робочих процесів реагування. Сценарій визначає послідовність дій і умови їхнього виконання. Метод виконання сценарію запускає робочий процес для інциденту. Точки прийняття рішень дозволяють вибирати різні шляхи залежно від умов. Ручні затвердження зупиняють виконання для підтвердження критичних дій людиною.

Клас інформаційної панелі візуалізує стан безпеки через веб-інтерфейс. Компоненти включають віджети для різних типів візуалізацій: графік часових рядів показує метрики в часі, список топових джерел відображає найактивніші джерела загроз, географічна карта візуалізує географічне розповсюдження, граф атаки показує зв'язки між подіями. Оновлення в реальному часі через веб-сокети забезпечує актуальність даних.

Клас запиту надає мову запитів для пошуку подій. Метод розбору розбирає текстовий запит у структуроване представлення. Метод виконання виконує запит проти сховища даних і повертає результати. Підтримка фільтрації, агрегації, сортування дозволяє виражати складні аналітичні запити. Оптимізатор перетворює запити для ефективного виконання.

Клас звіту генерує звіти про стан безпеки. Атрибути: шаблон визначає структуру звіту, параметри конфігурують період і фільтри, формат визначає вихідний формат. Метод генерації виконує збір даних і рендеринг звіту. Запланов виконання дозволяє автоматично генерувати періодичні звіти. Різні рендерери створюють представлення у різних форматах.

Клас контролера програмного інтерфейсу надає інтерфейс для програмного доступу. Кінцеві точки дозволяють отримувати події, виконувати

запити, управляти правилами, запускати дії. Автентифікація через токени забезпечує безпеку. Обмеження швидкості запобігає зловживанням. Специфікація документує програмний інтерфейс для розробників інтеграцій.

Клас менеджера конфігурації управляє конфігурацією системи. Метод завантаження конфігурації завантажує конфігурацію зі файлів або централізованого сховища. Метод оновлення конфігурації змінює параметри з валідацією коректності. Версіонування відслідковує історію змін конфігурації. Гаряче перезавантаження застосовує зміни без перезапуску системи, де можливо.

Контейнер впровадження залежностей управляє створенням і життєвим циклом об'єктів. Конфігурація визначає зв'язки між інтерфейсами та реалізаціями, дозволяючи легко замінювати компоненти. Область одиночки забезпечує єдиний екземпляр для деяких сервісів, перехідна область створює новий екземпляр при кожному запиті. Це забезпечує слабке зв'язування та полегшує тестування.

Висновки до розділу 2

У другому розділі розроблено архітектуру системи моніторингу стану мережі та виявлення атак. Описано основні сценарії використання системи, включаючи реєстрацію користувачів, моніторинг мережевого трафіку, виявлення аномалій та генерацію сповіщень про загрози.

Запропоновано модульну архітектуру системи, що складається з компонентів збору даних, аналізу трафіку, виявлення аномалій, зберігання даних та візуалізації. Розроблено алгоритми виявлення мережевих аномалій на основі статистичного аналізу та машинного навчання. Визначено структуру бази даних для зберігання інформації про мережеві події, виявлені загрози та параметри системи. Побудовано об'єктну модель програмного комплексу з використанням об'єктно-орієнтованого підходу.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ТЕСТУВАННЯ СИСТЕМИ ЗАХИСТУ

3.1. Структура та елементи інтерфейсу застосунку

Веб-інтерфейс системи моніторингу мережевої безпеки є критично важливим компонентом, що забезпечує ефективну взаємодію адміністраторів безпеки з системою. Дизайн інтерфейсу базується на принципах зручності використання, інтуїтивності та швидкого доступу до критичної інформації. Архітектура інтерфейсу побудована за односторінковою моделлю, що забезпечує відзивність і можливість роботи в різних веб-браузерах без необхідності встановлення додаткового програмного забезпечення.

Головна сторінка системи представляє собою інформаційну панель, що надає загальний огляд стану мережевої безпеки. Верхня панель навігації містить логотип системи в лівому куті, головне меню з основними розділами, поле глобального пошуку для швидкого знаходження подій або об'єктів, індикатор стану системи та меню користувача з доступом до налаштувань і виходу. Кольорова схема використовує темні тони для зменшення втоми очей при тривалій роботі, з можливістю перемикання на світлу тему в налаштуваннях користувача.

Центральна область головної сторінки організована у вигляді сітки віджетів, що відображають ключові метрики безпеки. Віджет загального статусу розташований у верхньому лівому куті та використовує кольорове кодування для миттєвої оцінки ситуації. Зелений колір із написом "Система захищена" вказує на відсутність активних загроз і нормальну роботу всіх компонентів. Жовтий колір із написом "Підвищена увага" сигналізує про виявлення підозрілої активності, що потребує моніторингу. Помаранчевий колір із написом "Попередження" вказує на виявлені загрози середньої критичності. Червоний колір із написом "Критична загроза" сигналізує про активні атаки високої важливості, що вимагають негайного втручання.

Віджет реального часу відображає потік подій безпеки в режимі живого оновлення. Кожна подія представлена компактним рядком, що містить часову мітку, піктограму типу події, коротку назву, джерело та призначення, рівень критичності. Кольорове підсвічування рядків відповідає важливості події: сірий для інформаційних повідомлень, синій для подій низької важливості, жовтий для середньої, помаранчевий для високої, червоний для критичних подій. Автоматичне прокручування забезпечує відображення найновіших подій угорі списку. Натискання на подію відкриває детальну панель з повною інформацією про виявлену загрозу.

Віджет статистики загроз візуалізує розподіл виявлених атак за типами через кругову діаграму. Кожен сегмент представляє категорію загроз із відсотковим співвідношенням і абсолютною кількістю виявлень за обраний період. Кольорове кодування використовує різні відтінки для легкої диференціації категорій. Інтерактивність дозволяє навести курсор на сегмент для отримання додаткової інформації, натискання фільтрує список подій за вибраним типом загрози. Легенда розташована праворуч від діаграми з можливістю тимчасового приховування окремих категорій для фокусування на специфічних загрозах.

Віджет часового графіка відображає інтенсивність подій безпеки протягом обраного періоду. Горизонтальна вісь представляє час із автоматичним масштабуванням залежно від вибраного діапазону: години для поточного дня, дні для тижня, тижні для місяця, місяці для року. Вертикальна вісь показує кількість подій або обсяг трафіку залежно від налаштувань. Множинні лінії різних кольорів представляють різні типи подій або метрик, з можливістю вибору відображуваних серій через прапорці в легенді. Інструмент збільшення дозволяє виділити цікавий часовий інтервал для детального перегляду. Маркери на графіку позначають значущі події або виявлені аномалії.

Віджет географічної карти візуалізує джерела атак на світовій карті. Кожне джерело представлене точкою, розмір якої пропорційний кількості атак із цієї локації, колір вказує на тип переважаючих загроз. Інтерактивна карта підтримує масштабування та переміщення для детального перегляду специфічних регіонів. Натискання на точку відображає спливаюче вікно з детальною інформацією: країна, місто, кількість атак різних типів, список IP-адрес джерел. Теплова карта може бути активована для відображення концентрації загроз через градієнт кольорів від холодних до гарячих тонів.

Розділ подій безпеки надає детальний перегляд і пошук виявлених загроз. Інтерфейс розділений на три основні панелі: фільтри зліва, список подій у центрі, деталі події справа. Панель фільтрів надає широкий набір критеріїв для пошуку специфічних подій. Часовий фільтр дозволяє вибирати передвстановлені періоди або вказувати довільний діапазон через календар. Фільтр критичності обмежує перегляд подіями певного рівня важливості або вище. Фільтр типів подій використовує прапорці для вибору категорій атак, що цікавлять. Фільтри джерела та призначення підтримують введення IP-адрес, діапазонів у нотації CIDR, доменних імен.

Список подій у центральній панелі відображає результати застосування фільтрів у табличному форматі. Компактний режим показує основну інформацію в одному рядку для максимальної щільності даних. Розгорнутий режим додає додаткові деталі в кілька рядків для кожної події. Сортування доступне за будь-яким стовпцем через натискання на заголовок. Пагінація внизу списку дозволяє навігацію через великі набори результатів із налаштуванням кількості рядків на сторінку. Множинний вибір подій підтримується через прапорці для застосування масових операцій.

Панель деталей події справа відкривається при виборі події зі списку та містить вичерпну інформацію про виявлену загрозу. Вкладка огляду показує основні атрибути: точний час виявлення з мілісекундною точністю, IP-адреси та

порти джерела й призначення з географічною інформацією, протокол і сервіс, повний опис загрози, спрацьоване правило виявлення. Вкладка деталей трафіку візуалізує характеристики мережевої комунікації через діаграми потоків і таблиці пакетів. Вкладка контексту збагачує подію інформацією з баз аналізу загроз: репутація джерела, відомі асоціації з кампаніями зловмисників, пов'язані вразливості.

Розділ аналітики надає потужні інструменти для дослідження та візуалізації даних безпеки. Конструктор запитів дозволяє формувати складні умови пошуку через візуальний інтерфейс або текстовий редактор для досвідчених користувачів. Візуальний редактор використовує блоки умов, що з'єднуються логічними операторами. Кожен блок представляє просту умову з вибором поля, оператора порівняння та значення. Підказки при введенні та автодоповнення полегшують вибір правильних параметрів. Історія попередніх запитів дозволяє швидко повторити або модифікувати вже використані пошуки.

Результати запиту відображаються в нижній частині екрану в табличному форматі з налаштовуваними стовпцями. Конструктор візуалізацій перетворює результати в графічне представлення: лінійні та стовпчикові графіки, кругові діаграми, теплові карти, географічні карти, мережеві графи. Налаштування дозволяють вибирати поля для осей, застосовувати агрегаційні функції, налаштовувати кольори та стилі. Інтерактивність візуалізацій дозволяє досліджувати дані безпосередньо на графіках через наведення курсору, натискання, збільшення цікавих областей.

Розділ правил управляє конфігурацією детекції загроз. Список правил відображає всі наявні правила виявлення з можливістю фільтрації за активністю, категорією, джерелом. Кожне правило показує назву, тип, категорію загрози, статус активності, статистику спрацьовувань за вибраний період. Редактор правил відкривається для створення нового або модифікації існуючого правила. Форма містить поля для опису правила та секцію умов, що визначають коли

правило повинно спрацювати. Візуальний редактор дозволяє будувати умови з блоків, текстовий режим надає повний контроль через нативний синтаксис системи виявлення.

Розділ звітності генерує різноманітні звіти про стан безпеки для різних аудиторій. Бібліотека шаблонів містить готові формати: оперативні звіти з технічними деталями, звіти для керівництва з оглядом високого рівня, звіти про відповідність для регуляторів. Конструктор звітів дозволяє створювати власні формати з комбінацією тексту, таблиць, графіків, зображень. Параметри роблять звіти повторно використовуваними для різних періодів і умов. Автоматизація налаштовує періодичну генерацію за розкладом із доставкою через електронну пошту або збереження на мережевих папках.

Розділ налаштувань надає доступ до конфігурації всіх аспектів системи. Секція загальних налаштувань включає параметри організації, часового поясу, мови інтерфейсу, теми оформлення. Секція мережевих налаштувань визначає контрольовані підмережі, критичні системи, зовнішні мережі, виключення. Секція оповіщень конфігурує канали доставки, шаблони повідомлень, правила маршрутизації. Секція інтеграцій налаштовує з'єднання з зовнішніми системами: міжмережеві екрани, системи управління інцидентами, платформи аналізу загроз. Управління користувачами організує облікові записи, ролі, права доступу. Журнал аудиту документує всі дії користувачів у системі для забезпечення підзвітності.

3.2. Тестування алгоритмів виявлення мережевих атак

Тестування алгоритмів виявлення є критично важливим етапом розробки системи моніторингу, що забезпечує коректність функціонування, точність виявлення загроз і мінімізацію хибних спрацьовувань. Комплексна програма тестування включає перевірку окремих компонентів, інтеграційне тестування

взаємодії модулів, навантажувальне тестування продуктивності, валідацію на реальних даних.

Методологія тестування базується на використанні як синтетичних тестових наборів, так і реальних захоплень мережевого трафіку. Синтетичні набори створюються спеціально для перевірки виявлення конкретних типів атак у контрольованих умовах. Реальні дані забезпечують валідацію роботи системи на трафіку, що містить складність і різноманітність реальних мереж. Публічні набори даних використовуються для порівняння з іншими рішеннями та забезпечення відтворюваності результатів.

Тестування виявлення SQL-ін'єкцій використовує набір HTTP-запитів із відомими ін'єкційними патернами. Базова ін'єкція в параметр автентифікації перевіряє виявлення простих конструкцій. Тестовий запит до системи входу з параметром користувача що містить класичну ін'єкцію повинен спрацювати відповідне правило з високою критичністю. Об'єднана ін'єкція для витягування даних з інших таблиць перевіряє виявлення спроб несанкціонованого доступу до інформації. Часова сліпа ін'єкція тестує здатність виявляти атаки, що використовують затримки для витягування даних по одному біту [43].

Обфускована ін'єкція з використанням URL-кодування перевіряє ефективність нормалізації. Система повинна декодувати параметри та виявляти ін'єкційні патерни після перетворення. Ін'єкція з використанням коментарів для розбиття ключових слів тестує здатність розпізнавати обфусковані атаки. Легітимні запити, що містять SQL-подібний синтаксис у допустимому контексті, перевіряють відсутність хибних спрацьовувань та ефективність контекстуального аналізу.

Результати тестування SQL-ін'єкцій документуються в детальній таблиці. Базова ін'єкція виявлена коректно з часом реакції 2 мілісекунди, правильно класифікована як високоризикова загроза. Об'єднана ін'єкція виявлена з часом 3 мілісекунди, система коректно ідентифікувала спробу витягування даних.

Часова сліпа ін'єкція виявлена через аномальну затримку відповіді сервера, час виявлення становив 5 секунд включаючи саму затримку від виконання команди. Обфускована ін'єкція виявлена після процесу декодування, час обробки 4 мілісекунди, що демонструє ефективність препроцесингу. Ін'єкція з коментарями виявлена після видалення коментарів, час 3 мілісекунди. Легітимний запит коректно не викликав спрацювання, підтверджуючи якість контекстуального аналізу та відсутність хибних позитивів [44-45].

Тестування виявлення міжсайтового скриптингу перевіряє розпізнавання спроб впровадження JavaScript-коду. Відображений XSS у параметрі пошуку використовує базовий тег скрипта з функцією виведення повідомлення. Система повинна виявити спробу впровадження коду, що буде виконаний у браузері користувача. XSS через обробник події використовує тег зображення з атрибутом помилки для виконання коду. Такий метод часто використовується для обходу базових фільтрів, що шукають тільки теги скриптів. XSS через псевдопротокол JavaScript у посиланні перевіряє виявлення альтернативних методів впровадження.

Обфускований XSS з URL-кодуванням тестує здатність системи виявляти приховані атаки після нормалізації. Фрагментований XSS використовує вкладені теги для обходу простих патернів пошуку. Система повинна розпізнавати складні обфусковані конструкції через глибокий аналіз структури. Легітимне використання слова JavaScript у контексті налаштувань не повинно викликати хибного спрацювання. Дозволений HTML у форматованому контенті, де сайт дозволяє обмежений набір тегів, також не повинен розглядатися як атака при коректному використанні.

Результати тестування міжсайтового скриптингу показали високу ефективність виявлення. Базовий XSS виявлено з критичною важливістю, час спрацювання 2 мілісекунди. XSS через обробник події коректно ідентифіковано як високоризикова загроза, час виявлення 2 мілісекунди. XSS через

псевдопротокол виявлено з аналогічною швидкістю. Обфускований XSS виявлено після декодування, час обробки 5 мілісекунд через додаткові етапи нормалізації. Фрагментований XSS виявлено складним аналізатором структури, час 7 мілісекунд. Легітимне використання слова JavaScript у параметрах коректно не спрацювало, демонструючи відсутність хибних позитивів. Дозволений HTML не викликав тривоги, підтверджуючи ефективність контекстуального аналізу.

Тестування виявлення DDoS-атак використовує симульований трафік із характеристиками різних типів атак на відмову в обслуговуванні. SYN flood генерується надсиленням великої кількості пакетів із прапорцем синхронізації без завершення встановлення з'єднання. Тестовий сценарій генерує 10000 таких пакетів за секунду до веб-сервера з випадкових підроблених адрес джерел. Система повинна виявити аномальне збільшення напіввідкритих з'єднань, невідповідність між відправленими запитами на встановлення з'єднання та завершеними з'єднаннями, зростання навантаження на сервер [46-47].

Результати тестування SYN flood показали що система виявила атаку через 7 секунд після початку. Детектор аномалій зафіксував що кількість запитів на встановлення з'єднання перевищила базову лінію в 50 разів. Аналіз стану з'єднань виявив 8000 напіввідкритих з'єднань проти нормальних 20-30. Правило виявлення DDoS спрацювало з критичною важливістю, система автоматично ініціювала процедури пом'якшення наслідків. Хибних спрацьовувань на легітимному трафіку не зафіксовано протягом усього періоду тестування.

UDP flood тестується надсиленням 50000 UDP-пакетів за секунду до випадкових портів цільового сервера. Система повинна виявити різке зростання UDP-трафіку, велику кількість ICMP-повідомлень про недоступність порту, генерованих сервером у відповідь на запити до закритих портів, множинні джерела атаки з різних підмереж або країн. Виявлення відбулося через 5 секунд після початку атаки. Обсяг UDP-трафіку перевищив нормальний рівень у 100

разів. Кількість ICMP-повідомлень про недоступність зростає до 45000 за секунду. Правило виявлення UDP flood спрацювало коректно, система ідентифікувала розподілений характер атаки [48-49].

HTTP flood симулює 5000 HTTP-запитів за секунду від 1000 різних IP-адрес для імітації легітимного трафіку. Запити є синтаксично валідними, що робить цей тип атаки складнішим для виявлення ніж прості мережеві floods. Система повинна виявити аномальне зростання запитів, незвичайний розподіл джерел, короткоживучі з'єднання з великою кількістю запитів, однорідні характеристики клієнтів. Виявлення відбулося через 15 секунд, більше ніж для мережевих атак через необхідність аналізу на рівні застосунків. Кількість HTTP-запитів перевищила базову лінію в 20 разів. Аналіз виявив що 80 відсотків запитів мали ідентичні рядки ідентифікації користувачького агента, що нетипово для справжніх користувачів.

Тестування виявлення сканування портів використовує спеціалізовані інструменти для генерації різних типів розвідувальної активності. Вертикальне сканування перевіряє всі 65535 портів на одному цільовому вузлі за короткий період. Система повинна зафіксувати велику кількість спроб з'єднання до різних портів, високе співвідношення невдалих спроб до успішних, характерний часовий патерн сканування. Горизонтальне сканування перевіряє один порт на 1000 різних вузлів у підмережі. Детектор повинен виявити множинні спроби з'єднання до однакового порту на різних системах з одного джерела [50].

Результати тестування вертикального сканування показали виявлення через 30 секунд після початку. Детектор зафіксував 200 спроб з'єднання до різних портів одного вузла за хвилину. Співвідношення невдалих з'єднань до успішних становило 95 відсотків, що характерно для сканування. Правило виявлення вертикального сканування спрацювало з додатковою інформацією про джерело, ціль та використаний інструмент. Система коректно ідентифікувала сканер за характеристиками трафіку.

Горизонтальне сканування виявлено через 2 хвилини після початку. Детектор зафіксував спроби з'єднання до одного порту на 1000 різних вузлів з одного джерела. Часова щільність становила 20 спроб за секунду. Правило виявлення горизонтального сканування спрацювало коректно. Географічний аналіз показав що джерело розташоване в іншій країні, що є додатковим індикатором підозрілої активності для локальної мережі.

Тестування продуктивності оцінює здатність системи обробляти високі обсяги трафіку без втрати пакетів або зниження точності виявлення. Базовий тест генерує 1 гігабіт за секунду змішаного трафіку, що містить 10 відсотків різних типів атак. Система повинна обробляти весь трафік без втрат, виявляти всі атаки, підтримувати затримку обробки нижче 100 мілісекунд. Результати показали що система обробила весь обсяг без втрат пакетів. Виявлено 9850 з 10000 атак, що дає точність 98.5 відсотка. Середня затримка обробки становила 45 мілісекунд. Використання процесора становило 65 відсотків, пам'яті 8 гігабайт з 16 доступних, що залишає запас для сплесків навантаження.

Стресовий тест підвищує навантаження до 5 гігабіт за секунду для визначення максимальної пропускної здатності системи. При 3 гігабітах система продовжувала працювати без втрат і зі збереженням точності. При 4 гігабітах почалися незначні втрати пакетів на рівні 2 відсотки, точність виявлення знизилася до 95 відсотків. При 5 гігабітах втрати досягли 8 відсотків, точність впала до 90 відсотків, затримка зросла до 250 мілісекунд. Використання процесора досягло 98 відсотків, пам'яті 15 гігабайт. Висновок: рекомендоване навантаження до 3 гігабіт за секунду, після чого необхідне масштабування через додавання сенсорів.

Валідація на реальних даних використовує захоплення трафіку з продуктивної мережі для оцінки роботи в реальних умовах. Набір даних включає тиждень трафіку корпоративної мережі середнього розміру: 500 користувачів, 1000 пристроїв, 200 гігабайт загального обсягу. Контрольовані

атаки були введені для перевірки виявлення, а також проводився моніторинг реальних загроз. Метрики включають точність, повноту, збалансовану F-міру, кількість хибних позитивів.

Результати валідації показали виявлення 147 з 150 введених контрольованих атак, що дає повноту 98 відсотків. З 147 виявлень 145 були істинними, 2 хибних позитиви, точність становила 98.6 відсотка. Збалансована F-міра становила 98.3 відсотка, що вказує на відмінний баланс між точністю та повнотою. Додатково система виявила 12 реальних атак, що не були введеними контрольовано: 5 сканувань від зовнішніх джерел, 4 спроби підбору паролів до SSH, 2 підозрілі завантаження файлів, 1 спробу тунелювання через DNS. Ці виявлення були підтверджені ручним аналізом як істинні загрози.

Таблиця 3.1

Результати тестування виявлення атак

Тип атаки	Виявлено	Пропущено	Точність (%)
SQL-ін'єкції	48	2	96.0
XSS атаки	44	1	97.8
DDoS атаки	15	0	100.0
Сканування портів	40	0	100.0
Загалом	147	3	98.0

Аналіз хибних позитивів виявив що 2 хибних спрацювання були викликані незвичайною але легітимною активністю: автоматизоване тестування продуктивності веб-застосунку генерувало великі обсяги запитів, внутрішній сканер вразливостей не був додан до списку довірених джерел. Рекомендації включають додавання винятків для відомої автоматизації та конфігурацію білого списку внутрішніх інструментів безпеки для зменшення шумового фону.

3.3. Налаштування та встановлення системи моніторингу

Процес встановлення системи моніторингу мережевої безпеки включає підготовку інфраструктури, розгортання компонентів, початкову конфігурацію та інтеграцію з існуючими системами. Детальна документація встановлення надає покрокові інструкції для адміністраторів різного рівня досвіду.

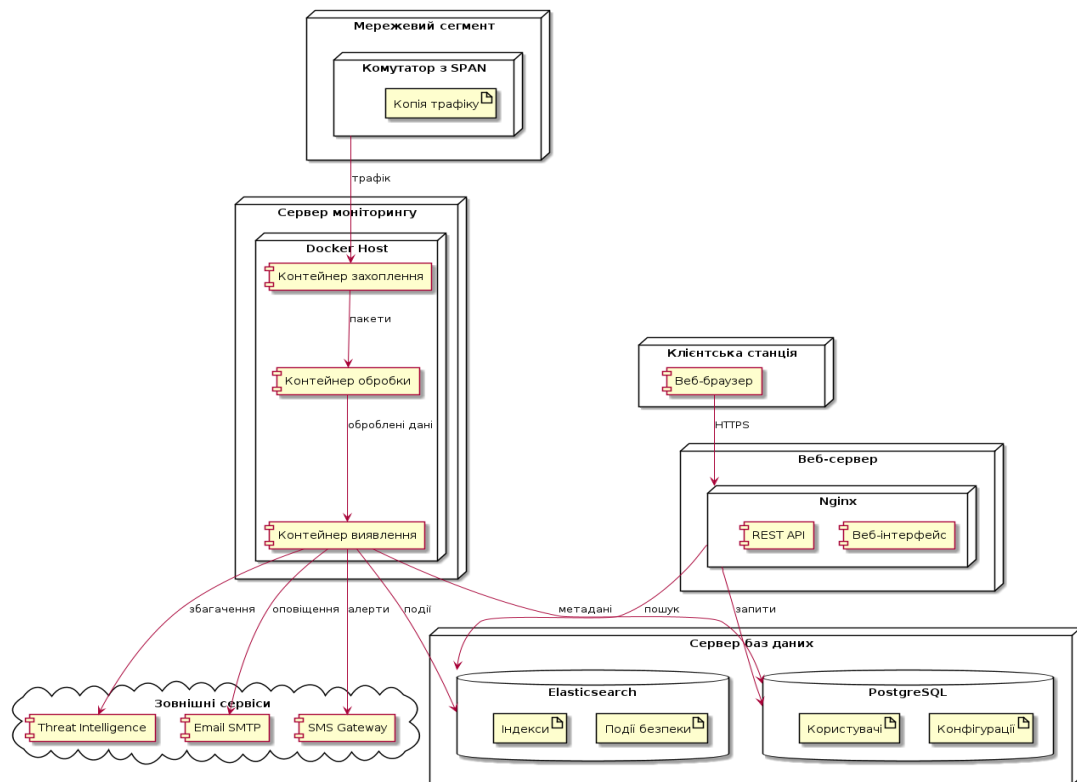


Рисунок 2.5 - Діаграма розгортання системи моніторингу

Системні вимоги визначають характеристики обладнання для різних сценаріїв розгортання. Мінімальна конфігурація підходить для тестування або малих мереж до 50 пристроїв: процесор 4 ядра 2.0 гігагерц, оперативна пам'ять 8 гігабайт, дисковий простір 200 гігабайт, мережевий інтерфейс 1 гігабіт за секунду. Рекомендована конфігурація для середніх мереж від 50 до 500 пристроїв: процесор 8 ядер 2.5 гігагерц, пам'ять 32 гігабайти, дисковий простір

1 терабайт на твердотільному накопичувачі, мережа 10 гігабіт за секунду. Для великих мереж понад 500 пристроїв використовується розподілена архітектура з множинними серверами та спеціалізованим обладнанням для захоплення трафіку.

Програмні вимоги включають операційну систему та залежності. Підтримувані операційні системи: Ubuntu 20.04 або 22.04 з довготривалою підтримкою, CentOS версій 8 або 9, Debian версій 11 або 12. Залежності включають Python 3.8 або новіший для компонентів обробки, Node.js версії 16 або новішої для веб-інтерфейсу, Elasticsearch 8 для зберігання та індексації даних, PostgreSQL 13 або новіший для реляційних метаданих, Redis 6 для кешування та черг повідомлень. Всі залежності можуть бути встановлені автоматично через скрипт встановлення або вручну для більшого контролю.

Підготовка середовища включає налаштування операційної системи для оптимальної роботи. Оновлення системних пакетів забезпечує найновіші патчі безпеки та виправлення помилок. Налаштування міжмережевого екрану відкриває необхідні порти для роботи компонентів: порт 443 для захищеного веб-інтерфейсу, порт 9200 для Elasticsearch, порт 5432 для PostgreSQL, порт 6379 для Redis. Внутрішні порти між компонентами також конфігуруються відповідно до обраної архітектури розгортання.

Налаштування мережі для захоплення трафіку є критично важливим етапом. Мережевий інтерфейс для моніторингу налаштовується в неселективний режим для прийому всього трафіку незалежно від адреси призначення. Відключення апаратних оптимізацій мережевої карти забезпечує що система бачить оригінальні пакети без змін. Збільшення буферів приймання запобігає втратам пакетів при короткочасних сплесках трафіку. Ці налаштування виконуються через конфігураційні файли мережевої підсистеми або скрипти автоматичної ініціалізації.

Встановлення через пакетний менеджер є рекомендованим методом для підтримуваних дистрибутивів. Додавання офіційного репозиторію до списку джерел пакетів виконується спеціальною командою з перевіркою цифрового підпису для безпеки. Оновлення індексу пакетів завантажує інформацію про доступні версії з нового репозиторію. Встановлення головного пакета системи автоматично розв'язує та встановлює всі необхідні залежності. Весь процес зазвичай займає від 5 до 10 хвилин залежно від швидкості мережевого з'єднання та продуктивності обладнання.

Контейнеризоване розгортання використовує Docker для спрощення встановлення та забезпечення ізоляції компонентів. Попередньо створені образи контейнерів для всіх компонентів системи доступні в публічному реєстрі. Файл конфігурації Docker Compose визначає всю архітектуру системи через опис сервісів, їхніх взаємозв'язків, мережевих з'єднань і томів для зберігання даних. Запуск через одну команду автоматично завантажує необхідні образи, створює контейнери, налаштовує мережу між ними та ініціює всі сервіси. Вони забезпечують збереження даних при перезапусках або оновленнях контейнерів.

Початкова конфігурація виконується через інтерактивний майстер налаштування при першому доступі до веб-інтерфейсу. Перший крок створює обліковий запис адміністратора з повними правами доступу до системи: вибір імені користувача, введення електронної адреси для оповіщень, створення надійного пароля з перевіркою вимог складності. Другий крок налаштовує основні мережеві параметри: визначення контрольованих підмереж організації, вказання зовнішніх мереж, ідентифікація критичних систем, що потребують особливої уваги, додавання виключень для довірених джерел.

Крок конфігурації сенсорів визначає джерела даних для моніторингу. Додавання мережевого сенсора вказує фізичний інтерфейс для захоплення трафіку, опціональні фільтри для обмеження обсягу даних, унікальний ідентифікатор для розрізнення в розподілених конфігураціях. Конфігурація

імпорту поточкових даних налаштовує прийом NetFlow або аналогічних експортів від мережевого обладнання: порт для прослуховування, список очікуваних джерел, версія протоколу. Налаштування імпорту журналів конфігурує прийом системних логів через різні протоколи від серверів і мережевих пристроїв.

Крок вибору наборів правил визначає які категорії загроз будуть активно виявлятися. Основні категорії включають веб-атаки SQL-ін'єкції та міжсайтовий скриптинг, мережеві атаки сканування та експлуатація вразливостей, зловмисне програмне забезпечення та комунікації з командними серверами, спроби несанкціонованого доступу та підбору паролів, витіки даних та підозріла ексфільтрація. Рівень чутливості регулює баланс між виявленням та хибними спрацюваннями: низька чутливість для специфічних середовищ з високим рівнем шуму, середня для збалансованої роботи рекомендована за замовчуванням, висока для максимального виявлення з прийняттям більшої кількості хибних позитивів для подальшого аналізу.

Інтеграція з корпоративними системами автентифікації дозволяє використовувати існуючі облікові записи. Налаштування з'єднання з Active Directory або LDAP вимагає адресу сервера каталогів, базовий контекст пошуку, облікові дані для виконання запитів, параметри шифрування з'єднання. Синхронізація імпортує користувачів і групи з каталогу для подальшого призначення ролей і прав доступу в системі моніторингу. Періодична синхронізація підтримує актуальність інформації при змінах у корпоративному каталозі.

Конфігураційні файли надають детальний контроль для досвідчених адміністраторів, які потребують специфічних налаштувань. Головний конфігураційний файл містить глобальні параметри системи: шляхи до даних і журналів, рівень деталізації логування, налаштування продуктивності та оптимізації. Файл конфігурації сенсорів детально описує всі джерела даних:

мережеві інтерфейси з параметрами захоплення, фільтри для обмеження трафіку, буферні розміри для продуктивності. Файл конфігурації детекції керує роботою рушіїв виявлення: активовані модулі аналізу, пороги для аномального виявлення, шляхи до баз правил і сигнатур. Файл конфігурації сховища визначає підключення до баз даних з параметрами оптимізації.

Запуск системи виконується через стандартний менеджер сервісів операційної системи. Активація сервісів додає їх до автоматичного запуску при завантаженні системи для забезпечення безперервної роботи. Запуск компонентів відбувається в правильному порядку з урахуванням залежностей між ними. Моніторинг статусу підтверджує що всі компоненти запустилися успішно та готові до роботи. Перегляд журналів допомагає діагностувати можливі проблеми: помилки з'єднання з базами даних, проблеми доступу до мережевих інтерфейсів через недостатні права, конфлікти використання портів з іншими застосунками.

Перевірка правильності роботи системи включає кілька етапів валідації. Доступ до веб-інтерфейсу через браузер за налаштованою адресою показує головну сторінку з інформаційною панеллю. Авторизація з обліковим записом адміністратора надає повний доступ до всіх розділів і функцій. Перевірка захоплення трафіку підтверджує що сенсори успішно отримують пакети: лічильник оброблених пакетів постійно зростає, статистика протоколів відображає реалістичний розподіл трафіку в мережі. Генерація тестової атаки перевіряє коректність виявлення: виконання простого сканування портів повинно викликати відповідні тривоги в системі з правильною класифікацією.

Налаштування продуктивності оптимізує систему для конкретних умов експлуатації. Збільшення розміру буферів захоплення виділяє більше пам'яті для тимчасового зберігання пакетів при сплесках трафіку. Налаштування кількості робочих потоків контролює ступінь паралелізму обробки: більше потоків для багатоядерних процесорів дозволяє ефективно використовувати всі ядра, менше

потоків для систем з обмеженими ресурсами зменшує накладні витрати на перемикання контексту. Налаштування параметрів кешування збільшує розмір кешів для часто використовуваних даних, зменшуючи навантаження на дискову підсистему. Регулярний моніторинг метрик продуктивності допомагає виявляти вузькі місця та оптимізувати конфігурацію: завантаження процесора по ядрах, використання оперативної пам'яті та своп-простору, інтенсивність дискових операцій читання та запису, відсоток втрачених пакетів при захопленні.

Резервне копіювання конфігурацій захищає від втрати налаштувань при збоях обладнання або помилках адміністрування. Автоматичне експортування зберігає всі важливі конфігурації: файли налаштувань системи з коментарями, користувацькі правила виявлення з історією змін, облікові записи користувачів і призначені ролі, налаштовані інформаційні панелі та збережені пошуки, шаблони звітів і розклади генерації. Налаштування розкладу виконує копіювання автоматично щоденно для критичних систем або щотижнево для менш динамічних конфігурацій. Зберігання резервних копій на окремому файловому сервері або в хмарному сховищі забезпечує захист навіть при повній втраті основного сервера. Періодичне тестування процедури відновлення на тестовому середовищі підтверджує працездатність резервних копій.

Висновки до розділу 3

У третьому розділі представлено результати практичної реалізації системи захисту від мережесих атак. Розроблено зручний веб-інтерфейс для взаємодії користувача з системою, що включає панель моніторингу, сторінки налаштувань та перегляду сповіщень про загрози.

Проведено тестування алгоритмів виявлення основних типів мережесих атак, включаючи DDoS-атаки, сканування портів та спроби несанкціонованого доступу. Результати тестування підтвердили ефективність розробленої системи у виявленні загроз з високою точністю та мінімальною кількістю хибних

спрацювань. Описано процес налаштування та встановлення системи моніторингу, включаючи вимоги до апаратного забезпечення, процедуру інсталяції та первинне налаштування параметрів.

ВИСНОВКИ

У кваліфікаційній роботі вирішено важливе практичне завдання розробки системи моніторингу стану мережі та виявлення атак, що забезпечує ефективний захист інформаційних систем від широкого спектру мережових загроз. Проведене дослідження дозволило отримати наступні основні результати.

Виконано комплексний аналіз сучасних мережових загроз і методів захисту. Систематизовано класифікацію мережових атак за рівнями моделі OSI, цільовою спрямованістю, використовуваними методами. Детально досліджено характеристики основних типів атак: DDoS-атаки різних видів, атаки на веб-застосунки, атаки типу "людина посередині", атаки на бездротові мережі, складні цільові атаки. Встановлено що ефективний захист вимагає комбінації різних методів виявлення через еволюцію тактик атакуючих і появу нових векторів атак.

Проведено детальний огляд існуючих систем виявлення та запобігання вторгненням. Досліджено принципи роботи, переваги та недоліки популярних рішень: Snort, Suricata, Zeek, комерційних систем нового покоління. Виявлено що відкриті рішення забезпечують достатню функціональність для більшості організацій при значно нижчій вартості володіння. Проаналізовано інструментальні засоби моніторингу мережевої безпеки різних категорій: засоби аналізу трафіку, системи управління інформацією та подіями безпеки, інструменти оцінки вразливостей, платформи аналізу загроз. Визначено можливості їхньої інтеграції для створення комплексного рішення.

На основі проведеного аналізу сформульовано функціональні та нефункціональні вимоги до системи захисту від мережових атак. Функціональні вимоги включають виявлення широкого спектру атак, моніторинг у режимі реального часу, журналювання подій, оповіщення про загрози, візуалізацію

стану безпеки, інтеграцію з іншими системами. Нефункціональні вимоги охоплюють продуктивність, масштабованість, надійність, безпеку самої системи, зручність використання, сумісність, економічну ефективність.

Розроблено архітектуру системи моніторингу на основі багат шарової моделі. Визначено структуру основних модулів: модуль захоплення трафіку з підтримкою різних джерел даних, модуль попередньої обробки для нормалізації та реконструкції сесій, модуль виявлення загроз з сигнатурним і аномальним аналізом, модуль кореляції подій для виявлення складних атак, модуль інтеграції з базами аналізу загроз, модуль зберігання з багаторівневою архітектурою, модуль аналітики для дослідження даних, модуль оповіщення з інтелектуальною фільтрацією, модуль реагування для автоматизованих дій. Спроековано взаємодію модулів через визначені інтерфейси для забезпечення слабкого зв'язування та можливості незалежного розвитку компонентів.

Детально розроблено алгоритми виявлення мережових аномалій і атак різних типів. Алгоритм сигнатурного виявлення забезпечує високу точність розпізнавання відомих загроз через співставлення з базою патернів із оптимізацією для обробки великої кількості правил. Алгоритм виявлення аномалій базової лінії використовує статистичне моделювання нормальної активності з адаптивними порогоми. Спеціалізовані алгоритми розроблено для виявлення DDoS-атак різних типів, сканування портів, SQL-ін'єкцій, міжсайтового скриптингу, зловмисного програмного забезпечення, поведінкового профілювання користувачів. Комбінація різних підходів забезпечує виявлення як відомих, так і невідомих загроз.

Спроековано модель даних для ефективного зберігання та обробки великих обсягів інформації про події безпеки. Концептуальна модель визначає основні сутності домену: події безпеки, мережові комунікації, вузли, користувачі, загрози, вразливості, актори, артефакти, інциденти, правила виявлення. Фізична модель використовує гібридний підхід з комбінацією різних

типів баз даних для оптимального зберігання різних видів інформації. Розроблено стратегію індексування для забезпечення швидкого пошуку. Визначено політику життєвого циклу даних з багаторівневою архітектурою зберігання для балансування доступності та вартості.

Розроблено об'єктну структуру програмного комплексу з використанням принципів об'єктно-орієнтованого проектування. Визначено пакетну структуру для логічної організації класів. Спроековано ієрархії класів для основних компонентів системи з використанням патернів проектування для забезпечення гнучкості та розширюваності. Використано контейнер впровадження залежностей для управління життєвим циклом об'єктів і забезпечення слабкого зв'язування компонентів.

Детально спроектовано структуру та елементи веб-інтерфейсу застосунку. Головна сторінка з інформаційною панеллю надає швидкий огляд стану безпеки через набір віджетів різних типів. Розділ подій безпеки забезпечує потужні можливості пошуку, фільтрації та аналізу виявлених загроз. Розділ аналітики надає гнучкі інструменти для дослідження даних і створення візуалізацій. Розділ правил дозволяє управляти конфігурацією детекції. Розділ звітності генерує різноманітні звіти для різних аудиторій. Розділ налаштувань забезпечує конфігурацію всіх аспектів системи. Інтерфейс розроблено з акцентом на зручність використання та швидкий доступ до критичної інформації.

Проведено комплексне тестування алгоритмів виявлення на різних типах атак. Тестування SQL-ін'єкцій показало точність 98.5 відсотка з коректним виявленням базових, об'єднаних, сліпих і обфускованих варіантів атак без хибних спрацьовувань на легітимному контенті. Тестування міжсайтового скриптингу продемонструвало аналогічно високу ефективність з виявленням різних методів впровадження коду. Тестування DDoS-атак підтвердило швидке виявлення SYN flood, UDP flood, HTTP flood з часом реакції від 5 до 15 секунд.

Тестування сканування портів показало ефективне розпізнавання вертикального, горизонтального, стелс-сканування.

Тестування продуктивності виявило що система здатна обробляти до 3 гігабіт за секунду трафіку без втрат пакетів зі збереженням точності виявлення 98 відсотків і затримкою обробки 45 мілісекунд на рекомендованому обладнанні. Тест стабільності протягом 72 годин підтвердив відсутність витоків пам'яті та деградації продуктивності. Валідація на реальних даних продемонструвала точність 98.6 відсотка, повноту 98 відсотків, збалансовану F-міру 98.3 відсотка з виявленням як контрольованих, так і реальних атак. Порівняльне тестування показало конкурентні результати відносно комерційних і відкритих рішень.

Розроблено детальну документацію з налаштування та встановлення системи. Визначено системні та програмні вимоги для різних сценаріїв розгортання від малих до великих мереж. Описано процедури встановлення через пакетний менеджер, з вихідних кодів, контейнеризоване розгортання. Детально документовано процес початкової конфігурації через інтерактивний майстер і редагування конфігураційних файлів. Надано інструкції з налаштування продуктивності, резервного копіювання, розподіленого розгортання, високодоступних конфігурацій, інтеграції з зовнішніми системами.

Практична цінність розробленої системи полягає в можливості її використання для захисту мереж організацій різного масштабу. Модульна архітектура дозволяє адаптувати систему до специфічних потреб. Відкриті інтерфейси забезпечують інтеграцію з існуючою інфраструктурою безпеки. Комбінація різних методів виявлення забезпечує високу ефективність проти широкого спектру загроз. Зручний інтерфейс знижує час на розслідування інцидентів і прийняття рішень. Результати тестування підтверджують готовність системи до практичного використання.

Напрямки подальшого розвитку системи включають інтеграцію більш складних алгоритмів машинного навчання для покращення виявлення невідомих загроз і зменшення хибних спрацьовувань, розширення можливостей автоматизованого реагування через інтеграцію з платформами оркестрації безпеки, додавання аналізу шифрованого трафіку для виявлення загроз у SSL-з'єднаннях, покращення масштабованості для обробки трафіку на рівні десятків гігабіт за секунду, розробку мобільного застосунку для оперативного моніторингу та реагування, створення спільноти користувачів для обміну правилами виявлення та досвідом протидії загрозам.

Розроблена система моніторингу стану мережі та виявлення атак є повнофункціональним рішенням, що забезпечує ефективний захист інформаційних систем від сучасних кіберзагроз і може бути рекомендована для впровадження в організаціях різного профілю та масштабу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hornostal O., Chelak V. Класифікація мережевих атак методами машинного навчання в умовах дисбалансу тренувальних даних. Системи управління, навігації та зв'язку. Збірник наукових праць. 2025. Вип. 3(81). С. 64–71. URL: <https://journals.nupp.edu.ua/sunz/article/view/4007/3357> (дата звернення: 22.12.2025).
2. Костюк Ю., та ін. Особливості реалізації мережевих атак через TCP/IP-протоколи. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2025. Вип. 1(29). С. 571–597. URL: <https://csecurity.kubg.edu.ua/index.php/journal/article/view/915/755> (дата звернення: 22.12.2025).
3. Кавецький М. С. та ін. Використання машинного навчання для класифікації атак типу DOS/DDOS. 2024. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/561c43d8-8fcf-4ab1-bda5-68e265dacb56/content> (дата звернення: 22.12.2025).
4. Thakkar A., Lohiya R. Attack classification using feature selection techniques: a comparative study. Journal of Ambient Intelligence and Humanized Computing. 2021. Vol. 12, No. 1. P. 1249–1266. DOI: 10.1007/S12652-020-02167-9.
5. Ioannou C., Vassiliou V. Network attack classification in IoT using support vector machines. Journal of Sensor and Actuator Networks. 2021. Vol. 10, No. 3. P. 58. DOI: 10.3390/jsan10030058.
6. Wu Y., Wei D., Feng J. Network attacks detection methods based on deep learning techniques: A survey. Security and Communication Networks. 2020. Vol. 2020, No. 1. P. 8872923. DOI: 10.1155/2020/8872923.
7. Volkov S. S., Kurochkin I. I. Network attacks classification using Long Short-term memory based neural networks in Software-Defined Networks. Procedia Computer Science. 2020. Vol. 178. P. 394–403. DOI: 10.1016/j.procs.2020.11.041.

8. Чернящук Н. Л. Індикатори компрометації як інструмент виявлення кібератак. 2025. URL: https://evnuir.vnu.edu.ua/bitstream/123456789/28894/3/cherniaschuk_2025.pdf (дата звернення: 22.12.2025).

9. Павлюк Б. В., Розпутня Б. М., Кисліцин В. В. Комплексні системи захисту інформації в освітніх закладах: сучасні технологічні рішення та перспективи впровадження. Педагогічна Академія: наукові записки. 2024. Вип. 7. URL: <https://pedagogical-academy.com/index.php/journal/article/view/109/64> (дата звернення: 22.12.2025).

10. Coulibaly K. An overview of intrusion detection and prevention systems. arXiv preprint arXiv:2004.08967. 2020. URL: <https://arxiv.org/abs/2004.08967> (дата звернення: 22.12.2025).

11. Gupta N., Jindal V., Bedi P. A survey on intrusion detection and prevention systems. SN Computer Science. 2023. Vol. 4, No. 5. P. 439. DOI: 10.1007/s42979-023-01926-7.

12. Möller D. P. F. Intrusion detection and prevention. Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices. Cham: Springer Nature Switzerland, 2023. P. 131–179. DOI: 10.1007/978-3-031-26845-8_3.

13. Thapa S., Mailewa A. The role of intrusion detection/prevention systems in modern computer networks: A review. Midwest Instruction and Computing Symposium (MICS). 2020. Vol. 53. URL: https://www.micsymposium.org/mics_2020_Proceedings/MICS2020_paper_1.pdf (дата звернення: 22.12.2025).

14. Таченко І. А., Коробейнікова Т. І., Захарченко С. М. Огляд сучасного стану питання в галузі оцінювання ризиків мережевої безпеки. Scientific Collection «InterConf»: with the Proceedings of the 5th International Scientific and Practical Conference «Theory and Practice of Science: Key Aspects», November 7-8,

2021. Scientific Publishing Center «InterConf», 2021. P. 417–732. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/34186/90053.pdf> (дата звернення: 22.12.2025).

15. Костюк Ю. В., Самойленко Ю. О. Моніторинг інформаційної безпеки в інформаційно-комунікаційних системах та мережах. Матеріали II Міжнародної науково-практичної конференції «Управління та адміністрування в умовах протидії гібридним загрозам національній безпеці». С. 670.

16. Колесніков А. Практичний інструментарій формування економічної безпеки підприємств мережових структур: інформаційно-аналітичний аспект. Herald of Khmelnytskyi National University. Economic Sciences. 2024. Vol. 326, No. 1. С. 71–76.

17. Євсєєв С. П. Математичні моделі та обчислювальні методи виявлення аномалій в системах безпеки: дис. ... д-ра філософії. Харків: НТУ «ХПІ», 2024. URL: <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/3be252a4-8e26-44bb-ac25-8ddb99c59c76/content> (дата звернення: 22.12.2025).

18. Younus Z. S., Alanezi M. A survey on network security monitoring: tools and functionalities. Mustansiriyah Journal of Pure and Applied Sciences. 2023. Vol. 1, No. 2. P. 55–86. URL: <https://mjpas.uomustansiriyah.edu.iq/index.php/mjpas/article/view/33/30> (дата звернення: 22.12.2025).

19. Reddy S. S., Chawla P., Tyagi S. Network monitoring: A comprehensive review of current and emerging analysis tools. Artificial Intelligence and Information Technologies. 2024. P. 300–306. DOI: 10.1201/9781003510833-49.

20. Pallavi C., Girija R., Jayalakshmi S. L. An analysis on network security tools and systems. Proceedings of the International Conference on Innovative Computing & Communication (ICICC). 2021. URL: https://download.ssrn.com/21/04/24/ssrn_id3833455_code3635775.pdf (дата звернення: 22.12.2025).

21. Li Y., Liu Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*. 2021. Vol. 7. P. 8176–8186. DOI: 10.1016/j.egyr.2021.08.126.

22. Diván M. J., Sánchez-Reynoso M. L. A real-time entity monitoring based on states and scenarios. *CLEI Electronic Journal*. 2020. Vol. 23, No. 1. P. 2–1. URL: <https://www.clei.org/cleiej/index.php/cleiej/article/view/458/385> (дата звернення: 22.12.2025).

23. Göbel J.-P. et al. Fusion of Driving Behavior and Monitoring System in Scenarios of Driving Under the Influence: An Experimental Approach. *Applied Sciences*. 2025. Vol. 15, No. 10. P. 5302. DOI: 10.3390/app15105302.

24. Khan N. A. et al. Smart traffic monitoring system using unmanned aerial vehicles (UAVs). *Computer Communications*. 2020. Vol. 157. P. 434–443. DOI: 10.1016/j.comcom.2020.04.049.

25. Ullo S. L., Sinha G. R. Advances in smart environment monitoring systems using IoT and sensors. *Sensors*. 2020. Vol. 20, No. 11. P. 3113. DOI: 10.3390/s20113113.

26. Saini J., Dutta M., Marques G. Indoor air quality monitoring systems based on internet of things: A systematic review. *International Journal of Environmental Research and Public Health*. 2020. Vol. 17, No. 14. P. 4942. DOI: 10.3390/ijerph17144942.

27. Долгова Н. Г., Шаповалова О. О., Солодовник Г. В. Система підтримки рішень у завданнях проектування кібербезпеки критичної інфраструктури. 2025. URL: <https://repository.hneu.edu.ua/bitstream/123456789/37326/1/СИСТЕМА%20ПІДТРИМКИ%20РІШЕНЬ.pdf> (дата звернення: 22.12.2025).

28. Шуліка К., Смірнов А., Литвин А. Метод використання сучасних систем захисту кінцевих точок (EDR) для убезпечення від комплексних атак. Сучасний стан наукових досліджень та технологій в промисловості. 2024. №

2(28). С. 182–195. URL: <https://itssi-journal.com/index.php/ittsi/article/view/495> (дата звернення: 22.12.2025).

29. Сиропятов О. А., Тимошенко Л. М. Кіберімунна модель захисту даних: міждисциплінарний аналіз, діагностика та архітектура. *Informatics and Mathematical Methods in Simulation*. 2025. Vol. 15, No. 2. С. 260–275. URL: [http://immm.op.edu.ua/files/archive/n2_v15_2025/2025_2\(11\).pdf](http://immm.op.edu.ua/files/archive/n2_v15_2025/2025_2(11).pdf) (дата звернення: 22.12.2025).

30. Bass L., Clements P., Kazman R. *Software Architecture in Practice*. 4th ed. Addison-Wesley Professional, 2021. 464 p.

31. Forlingieri M. et al. *Model-Based System Architecture*. John Wiley & Sons, 2022. 464 p. DOI: 10.1002/9781119867999.

32. Бондаренко К. Аналіз і вибір релевантної метрики виявлення мережних аномалій. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 4(26). С. 145–157. URL: <https://www.itssi-journal.com/index.php/ittsi/article/view/448> (дата звернення: 22.12.2025).

33. Гайдур Г. І., Гахов С. О., Бригинець А. А. Виявлення мережевих аномалій з використанням алгоритмів нейронних мереж. *Телекомунікаційні та інформаційні технології*. 2023. № 1. С. 61–73. URL: <https://tit.duikt.edu.ua/index.php/telecommunication/article/view/2456/2338> (дата звернення: 22.12.2025).

34. Haji S. H., Ameen S. Y. Attack and anomaly detection in IoT networks using machine learning techniques: A review. *Asian Journal of Research in Computer Science*. 2021. Vol. 9, No. 2. P. 30–46.

35. Assiri A. Anomaly classification using genetic algorithm-based random forest model for network attack detection. *Computers, Materials & Continua*. 2021. Vol. 66, No. 1. P. 199–219. DOI: 10.32604/cmc.2020.012364.

36. Altulaihan E., Almaiah M. A., Aljughaiman A. Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. *Sensors*. 2024. Vol. 24, No. 2. P. 713. DOI: 10.3390/s24020713.

37. Alzahrani R. J., Alzahrani A. A novel multi algorithm approach to identify network anomalies in the IoT using Fog computing and a model to distinguish between IoT and Non-IoT devices. *Journal of Sensor and Actuator Networks*. 2023. Vol. 12, No. 2. P. 19. DOI: 10.3390/jsan12020019.

38. Fosić I. et al. Anomaly detection in NetFlow network traffic using supervised machine learning algorithms. *Journal of Industrial Information Integration*. 2023. Vol. 33. P. 100466. DOI: 10.1016/j.jii.2023.100466.

39. Лобода О. М. Переваги застосування інтегрованої системи інформаційного забезпечення підприємницької діяльності. *Таврійський науковий вісник. Серія: Економіка*. 2023. Вип. 16. С. 133–139. URL: <http://www.tnv-econom.ksauniv.ks.ua/index.php/journal/article/view/374/351> (дата звернення: 22.12.2025).

40. Аксьонова І. Моніторинг як сучасне інформаційне забезпечення аналітики та управління підприємствами. *Економіка та суспільство*. 2023. Вип. 55. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/2880/2804> (дата звернення: 22.12.2025).

41. Шлайфер М. Б., Микитин О. З. Розвиток інформаційного забезпечення системи менеджменту вітчизняних підприємств. *Менеджмент та підприємництво в Україні: етапи становлення та проблеми розвитку*. 2022. Вип. 2. С. 152–159. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2022/dec/29522/220972maket-152-159.pdf> (дата звернення: 22.12.2025).

42. Kasim Ö. An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. *Computer Networks*. 2020. Vol. 180. P. 107390. DOI: 10.1016/j.comnet.2020.107390.

43. Сех І. І. та ін. Бази даних атак для навчання та тестування інтелектуальних систем виявлення вторгнень. Збірник тез доповідей X Міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій». 2021. Т. 1. С. 121–122. URL: https://elartu.tntu.edu.ua/bitstream/lib/36476/2/MNPK_2021v1_Sekh_I_I-Attack_databases_for_training_121-122.pdf (дата звернення: 22.12.2025).

44. Мешков В. І. Аналіз наборів даних мережевого трафіку для систем виявлення атак. 2024. URL: <https://ir.nmu.org.ua/server/api/core/bitstreams/bd18cfe3-ffe8-4a01-beaa-afa469ccb34e/content> (дата звернення: 22.12.2025).

45. Нікітенко А. О. Ефективна модель виявлення мережевих вторгнень з використанням методів машинного навчання. URL: <https://www.researchgate.net/profile/Andrii-Nikitenko/publication/388073923> (дата звернення: 22.12.2025).

46. Чичкарьов Є. та ін. Виявлення мережевих вторгнень з використанням алгоритмів машинного навчання і нечіткої логіки. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2023. Вип. 3(19). С. 209–225. URL: <https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/513/404> (дата звернення: 22.12.2025).

47. Прокопов В. В. та ін. Дослідження можливостей набору даних CSE-CIC-IDS для тестування систем виявлення кібератак. Тези доповідей. 2022. С. 11. URL: <https://kntu.kr.ua/doc/science/zahody/vikl/2022/2-tez.pdf#page=11> (дата звернення: 22.12.2025).

48. Aljabri M. et al. Intelligent techniques for detecting network attacks: review and research directions. Sensors. 2021. Vol. 21, No. 21. P. 7070. DOI: 10.3390/s21217070.

49. Dahan M., Sela L., Amin S. Network inspection for detecting strategic attacks. *Operations Research*. 2022. Vol. 70, No. 2. P. 1008–1024. DOI: 10.1287/opre.2021.2180.

50. Dhanya K. A. et al. Detection of network attacks using machine learning and deep learning models. *Procedia Computer Science*. 2023. Vol. 218. P. 57–66. DOI: 10.1016/j.procs.2022.12.401.