НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ВИКОРИСТАННЯ АВТОМАТИЧНИХ СИСТЕМ ЗБОРУ ІНФОРМАЦІЇ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ КОМП'ЮТЕРНОЇ МЕРЕЖІ»

Виконав: здобувач 4 курсу

Рівень бакалавр

Галузь знань 12 «Інформаційні технології»,

спеціальність 125 «Кібербезпека»

Стоянов Даніїл Олегович

Керівник: к.т.н., доцент

Бойко Віктор Дмитрович

Рецензент: к.т.н., доцент

Фразе-Фразенко Олексій Олексійович

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **кібербезпеки**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **125 Кібербезпека**
Назва освітньої програми: **Кібербезпека**

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
професор С.А. Горбаченко
_____ «____ »_____ 20__ року

З А В Д А Н Н Я
на кваліфікаційну (бакалаврську) роботу
здобувачу Стоянову Данііlу Олеговичу

1. Тема роботи: «Використання автоматичних систем збору інформації для підвищення безпеки комп'ютерних мереж»
Керівник роботи: к.т.н, доцент Бойко Віктор Дмитрович
затверджені наказом НУ ОЮА від «18» березня 2025 року №548-6
2. Строк подання здобувачем роботи «19» травня 2025 року
3. Дата видачі завдання «16» вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та пошук науковотехнічної інформації за темою роботи	Вересень-жовтень 2024	
2	Узгодження теми з науковим керівником, формулювання мети завдань дослідження	Листопад 2024	
3	Робота над першим розділом	Листопад-грудень 2024	
4	Робота над другим розділом	Січень-лютий 2025	
5	Робота над третім розділом	Лютий-березень 2025	
6	Уточнення подальшого обсягу роботи та його коригування	Березеньтравень 2025	
7	Оформлення звітної частини	Травень 2025	
8	Захист роботи	11.06.2025	

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Використання автоматичних систем збору інформації для підвищення безпеки комп’ютерної мережі» на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека, освітньою програмою: Кібербезпека, містить 58 сторінок, 9 рисунків та 20 джерел у переліку посилань.

Об’єктом дослідження є процес забезпечення безпеки комп’ютерної мережі.

Предметом дослідження є програмні засоби для автоматизованого збору та аналізу інформації з метою виявлення потенційних загроз у мережі.

Метою дослідження є підвищення ефективності виявлення кіберзагроз та аномалій у комп’ютерних мережах шляхом розробки програмного забезпечення, яке автоматизує процес збору та аналізу технічних даних з вузлів мережі та мережевого трафіку.

Розроблене рішення може бути використане адміністраторами інформаційної безпеки, фахівцями з кіберзахисту, а також викладачами у сфері IT-освіти.

Подальші напрями досліджень включають інтеграцію з системами SIEM, автоматизацію реагування на інциденти та реалізацію візуалізації зібраних даних у реальному часі.

АВТОМАТИЗОВАНИЙ ЗБІР ІНФОРМАЦІЇ, ARP-ТАБЛИЦЯ, МОНІТОРИНГ МЕРЕЖІ, КІБЕРБЕЗПЕКА, PYTHON, ВИЯВЛЕННЯ ВТОРГНЕНЬ, IP-АНАЛІЗ, ПРОЦЕСИ, ВІДКРИТІ ПОРТИ, GEOIP, WHOIS.

ABSTRACT

The bachelor's thesis titled "Using Automated Information Collection Systems to Enhance Computer Network Security", submitted for obtaining the first (bachelor's) level of higher education in the specialty 125 Cybersecurity, educational program: Cybersecurity, consists of 58 pages, 9 figures, and 20 sources included in the reference list.

The object of the study is the process of ensuring computer network security.

The subject of the study is software tools for automated data collection and analysis to detect potential network threats.

The purpose of this research is to increase the effectiveness of detecting cyber threats and anomalies in computer networks by developing software that automates the collection and analysis of technical data from network hosts and traffic.

The developed solution can be applied by information security administrators, cybersecurity specialists, and educators involved in IT training.

Future research directions include integration with SIEM systems, automation of incident response, and real-time visualization of collected data.

AUTOMATED INFORMATION COLLECTION, ARP TABLE, NETWORK MONITORING, CYBERSECURITY, PYTHON, INTRUSION DETECTION, IP ANALYSIS, PROCESSES, OPEN PORTS, GEOIP, WHOIS.

ЗМІСТ

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій питання забезпечення безпеки комп’ютерних мереж набуває особливої важливості. З кожним роком зростає кількість як користувачів мережевих ресурсів, так і кібератак, що спрямовані на несанкціонований доступ, модифікацію або знищення інформації. Особливої уваги потребують локальні мережі організацій, адже в них зосереджено значні обсяги конфіденційних даних, що становлять цінність як для самих організацій, так і для зловмисників.

Традиційні засоби захисту, такі як антивірусне програмне забезпечення та фаєрволи, вже не забезпечують належного рівня захисту без підтримки з боку сучасних систем моніторингу й аналізу трафіку. Саме тому все більшого поширення набувають автоматизовані системи збору інформації про мережеву активність, які дозволяють оперативно виявляти підозрілі дії та реагувати на інциденти безпеки.

Розробка власного програмного забезпечення, здатного виконувати базові функції моніторингу мережі та фіксації потенційних загроз, є актуальним напрямом, особливо для малих підприємств, що не мають змоги впроваджувати дорогі комерційні рішення.Метою даної бакалаврської роботи є розробка та дослідження програмного забезпечення для автоматичного збору інформації в комп’ютерній мережі з метою підвищення її безпеки шляхом виявлення потенційних загроз та аномальної активності.Для досягнення поставленої мети в роботі необхідно вирішити такі завдання:

- Проаналізувати існуючі методи та засоби забезпечення безпеки комп’ютерних мереж.
- Дослідити роль автоматичних систем збору інформації в системах моніторингу та реагування на інциденти.
- Сформувати вимоги до власного програмного забезпечення для збору та обробки мережевої інформації.
- Розробити архітектуру та реалізувати прототип ПЗ з використанням

сучасних технологій.

- Провести тестування розробленого програмного забезпечення в умовах мережевої активності.
- Оцінити ефективність роботи створеного засобу збору інформації та його можливості для подальшого вдосконалення.

Об’єктом дослідження є процеси забезпечення інформаційної безпеки комп’ютерних мереж.

Предметом дослідження є методи та програмні засоби автоматичного збору, аналізу й обробки інформації про мережеву активність для виявлення потенційних загроз та аномальної поведінки в мережі.

У процесі виконання роботи було використано такі методи дослідження:

- аналіз та узагальнення наукових джерел з тематики кібербезпеки, мережевого моніторингу та автоматизованих систем збору даних;
- порівняльний аналіз функціональних можливостей існуючих систем збору та обробки інформації в комп’ютерних мережах;
- моделювання архітектури програмного забезпечення;

експериментальні дослідження, що включають розробку, тестування і оцінку ефективності створеного програмного рішення в умовах симульованої мережевої активност

1 ТЕОРЕТИЧНІ ОСНОВИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОМП'ЮТЕРНИХ МЕРЕЖ

1.1 Основні принципи та поняття безпеки комп'ютерних мереж

Для початку визначимося що таке безпека комп'ютерних мереж, - це сукупність заходів, які спрямовані на захищеності інформаційних ресурсів та систем від різноманітних загроз, таких як несанкціонований доступ, зловмисне втручання, втрата або пошкодження даних. Принцип цих систем забезпечує їхню конфіденційність, цілісність і доступність до інформаційних ресурсів, які зберігаються, передаються або обробляються в комп'ютерних мережах. Безпека комп'ютерних мереж включає в себе не лише технологічні рішення, але й політичні процедури та організаційні заходи, які забезпечують рівень захисту.

Однак на практиці безпека комп'ютерних мереж не може бути забезпечена лише локальними або індивідуальними заходами. В умовах глобалізації та швидкого розвитку інформаційних технологій, мережі, що з'єднують організації та держави, потребують узгоджених підходів до забезпечення безпеки. Тому для досягнення єдиного рівня захисту в умовах міжнародної взаємодії та обміну даними, особливо в межах великих організацій і на глобальному рівні, виникає необхідність у розробці та впровадженні єдиних стандартів безпеки.

Міжнародні стандарти безпеки комп'ютерних мереж є важливим інструментом для визначення вимог до захисту інформації в різних країнах і організаціях. Вони забезпечують єдині норми та рекомандації, що реалізують забезпечення необхідного рівня захисту на всіх етапах передачі та обробки даних. Ці стандарти регулюють не лише технічні елементи безпеки, а й організаційна та управлінські заходи.

Найбільш поширеними стандартами безпеки є ISO/IEC 27001 та NIST SP 800-53. Розглянемо для початку стандарт безпеки ISO/IEC 27001 він визначає вимоги до системи управління інформаційною безпекою (ISMS). Цей стандарт охоплює всі аспекти захисту даних, включаючи ідентифікацію та

оцінку ризиків, встановлення політик безпеки, а також управління інцидентами безпеки.

Стандарт NIST SP 800-53, він був розроблений Національним інститутом стандартів і технологій США (NIST), забезпечує детальні рекомендації з безпеки інформаційних систем та мереж, включаючи впровадження заходів, спрямованих на доступ, моніторинг безпеки та захисту даних. Ці стандарти є основою для розробки політик безпеки в багатьох країнах і організаціях по всьому світу, забезпечуючи глобальну уніфікацію підходів до інформаційної безпеки.

Після розляду міжнародних стандартів безпеки, які визначають загальні вимоги до безпеки, важливо звернути уваги на основні принципи безпеки, що лежать в основі цих стандартів. Ці принципи безпеки формують основу для побудови ефективної системи захисту. Три ключових з них – конфіденційність, цілісність та доступність – утворюють так звану модель CIA (Confidentiality, Integrity, Availability. Ця модель є фундаментальною концепцією інформаційної безпеки, яка застосовується при проєктуванні, реалізації та оцінці захисних механізмів.

- Конфіденційність (Confidentiality) - забезпечення доступу до інформації лише уповноваженим особам. Це означає, що будь-які дані, які передаються через мережу, не повинні бути розкриті або прочитані третіми особами без дозволу.
- Цілісність (Integrity) - гарантія того, що дані не будуть змінені, знищені або підроблені без відповідного дозволу. Забезпечення цілісності передбачає виявлення будь-яких несанкціонованих змін інформації.
- Доступність (Availability) - забезпечення постійного та своєчасного доступу до інформації та ресурсів уповноваженим користувачам. Навіть найзахищеніша система не буде ефективною, якщо користувачі не зможуть використовувати її ресурси у потрібний момент.

Окрім базових принципів CIA, у сфері комп’ютерної безпеки застосовуються також такі поняття:

- Аутентифікація (Authentication) - підтвердження особи користувача або джерела даних.
- Авторизація (Authorization) - визначення прав доступу до ресурсів.
- Незаперечність (Non-repudiation) - гарантія того, що учасник взаємодії не зможе заперечити свою участь або авторство дії.
- Відповідальність (Accountability) - здатність відслідковувати дії користувачів та визначати, хто і коли здійснив ті чи інші операції.

Ці принципи є універсальними і застосовуються в більшості моделей забезпечення інформаційної безпеки. Вони лягають в основу розробки політик безпеки, архітектури захищених систем та вибору відповідних засобів захисту. Розуміння та реалізація цих принципів є ключовими елементами у створенні ефективної системи захисту комп'ютерної мережі, що дозволяє не лише знизити ризики, а й забезпечити стабільність функціонування інформаційної інфраструктури.

Політики безпеки відіграють ключову роль у практичній реалізації принципів комп'ютерної безпеки. Вони є документованими правилами, процедурами та стандартами, що регламентують поведінку користувачів і адміністраторів, а також порядок взаємодії з інформаційними ресурсами організації.

Головна мета політик безпеки - забезпечення узгодженості дій усіх учасників інформаційної системи, запобігання інцидентам безпеки та створення умов для ефективного реагування на потенційні загрози. Політика безпеки має враховувати специфіку діяльності організації, критичність даних, можливі загрози та відповідні механізми захисту.Серед основних завдань політик безпеки можна виділити:

- визначення прав доступу користувачів до інформаційних ресурсів;
- регламентування процесів автентифікації та авторизації;
- встановлення правил резервного копіювання та відновлення даних;
- визначення порядку моніторингу, ведення журналів подій та аналізу інцидентів;
- формулювання процедур реагування на порушення безпеки.

Ефективна політика безпеки також сприяє дотриманню вимог стандартів (зокрема ISO/IEC 27001) та законодавства у сфері захисту інформації, а її дотримання підвищує загальний рівень кіберзахисту організації.

Отже, політики безпеки є практичним інструментом втілення теоретичних принципів захисту - вони трансформують модель CIA та інші концепції у конкретні механізми, що забезпечують стійкість комп'ютерної мережі до внутрішніх і зовнішніх загроз.

1.2 Класифікація атак та методи захисту комп'ютерних мереж

Сучасні комп'ютерні мережі, які забезпечують критичні процеси в організаціях та повсякденному житті, постійно перебувають під загрозою з боку кіберзлочинців. З кожним роком кількість та складність мережевих атак зростає, що вимагає не лише виявлення нових вразливостей, а й розробки ефективних засобів захисту.

Для побудови надійної системи кібербезпеки необхідно чітко розуміти, які типи атак існують, як вони класифікуються за способом впливу та цілями, а також якими методами можна запобігти їх реалізації або мінімізувати наслідки. Класифікація атак дозволяє систематизувати знання про потенційні загрози, а відповідні захисні механізми забезпечують ефективне реагування на інциденти безпеки.

У цьому підрозділі буде розглянуто основні види атак на комп'ютерні мережі та методи їхньої протидії, зокрема засоби технічного, криптографічного та організаційного захисту.

Комп'ютерні мережі щоденно зазнають впливу різноманітних кіберзагроз, які можуть порушити конфіденційність, цілісність або доступність інформації. Для ефективного забезпечення безпеки інформаційних систем необхідно розуміти природу основних типів атак, їхні цілі та методи реалізації.

Атаки на доступність спрямовані на виведення з ладу або ускладнення доступу до мережевих ресурсів. Найпоширенішими є атаки типу DoS (Denial of

Service) та DDoS (Distributed Denial of Service), при яких зловмисники надсилають велику кількість запитів до цільового сервера, внаслідок чого той перестає відповідати на легітимні запити користувачів.

Ці атаки можуть використовувати різні протоколи: TCP SYN flood, UDP flood, HTTP flood тощо. Зазвичай вони реалізуються за допомогою ботнетів - мереж заражених пристроїв, керованих з однієї точки.

Цей тип атак націлений на несанкціонований доступ до чутливої інформації. Одним із найпоширеніших методів є перехоплення трафіку (packet sniffing), за якого зловмисник аналізує незашифровані пакети даних у мережі. Інший приклад - атака “людина посередині” (Man-in-the-Middle, MITM), при якій зловмисник перехоплює та змінює інформацію, що передається між двома сторонами, не викликаючи підозри.

Подібні атаки особливо небезпечні в незахищених або публічних мережах, де трафік не шифрується належним чином.

Атаки на цілісність інформації передбачають її зміну або підміну. Типовим прикладом є SQL-ін’єкції, коли зловмисник вводить шкідливий SQL-код через форму введення даних на вебсайті з метою отримання, змінення або видалення записів у базі даних.

До атак на цілісність також належать несанкціоновані зміни конфігураційних файлів, знищення логів, зміна критично важливих системних даних тощо.

У відповідь на зростання кількості кіберзагроз, організації впроваджують різноманітні технічні та організаційні заходи для забезпечення захисту своїх мережевих ресурсів. Методи захисту охоплюють як превентивні, так і реактивні заходи, що спрямовані на зменшення ризиків порушення конфіденційності, цілісності та доступності інформації.

Брандмауери (firewalls) є одним із найпоширеніших засобів захисту мережі. Вони контролюють вхідний та вихідний трафік, дозволяючи або блокуючи з’єднання згідно з визначеними правилами. Існують апаратні та програмні брандмауери, які можуть бути встановлені як на рівні мережі, так і на окремих вузлах.

Системи виявлення та запобігання вторгненням (IDS/IPS) забезпечують аналіз трафіку з метою виявлення підозрілої активності або вже відомих шаблонів атак.

- IDS (Intrusion Detection System) фіксує підозрілу активність і сповіщає адміністратора.
- IPS (Intrusion Prevention System), на відміну від IDS, може автоматично блокувати трафік у разі виявлення загрози.

Шифрування є ключовим методом забезпечення конфіденційності інформації під час її передавання або зберігання. Існують два основних типи шифрування:

- Симетричне шифрування (наприклад, AES) - використовує один ключ для шифрування та дешифрування.
- Асиметричне шифрування (наприклад, RSA) - використовує пару ключів: публічний і приватний.

Протоколи, такі як HTTPS, SSL/TLS, забезпечують захищене з’єднання між клієнтом і сервером, запобігаючи перехопленню чи зміні даних під час передачі.

Організаційні заходи безпеки, такі як контроль доступу, дозволяють обмежити доступ користувачів до мережевих ресурсів відповідно до їхніх ролей. До таких засобів належать:

- ACL (Access Control Lists) - дозволяють налаштувати правила доступу на основі IP-адрес, портів або протоколів.
- RBAC (Role-Based Access Control) - дозволяє призначати права на основі ролей користувачів у системі.
- Многофакторна аутентифікація (MFA) - додає додаткові рівні перевірки користувача, знижуючи ризики несанкціонованого доступу.

Вибір і впровадження засобів захисту комп’ютерних мереж повинні супроводжуватися регулярною оцінкою їхньої ефективності. Такий підхід дозволяє визначити, наскільки обрані механізми відповідають поточним загрозам, чи не

виникають нові вразливості, а також оптимізувати витрати на інформаційну безпеку. Ключові критерії оцінки:

- Виявлення загроз - ефективна система повинна мати здатність вчасно виявляти підозрілу активність. Це стосується як сигнатурного виявлення (IDS/IPS), так і поведінкового аналізу.
- Швидкість реагування - важливим показником є здатність системи швидко зреагувати на загрозу (автоматично чи вручну) і зменшити потенційні збитки.
- Простота адміністрування - методи, що потребують великого втручання або складні в налаштуванні, можуть спричиняти людські помилки або затримки.
- Масштабованість - засоби захисту повинні адаптуватися до зростання мережі та кількості користувачів без втрати ефективності.
- Відсутність хибних спрацювань - висока точність виявлення загроз без великої кількості false positives та false negatives.

Методи оцінювання:

- Аудит інформаційної безпеки - комплексна перевірка мережевої інфраструктури, політик доступу, конфігурацій.
- Пенетраційне тестування (penetration testing) - моделювання атак з метою виявлення вразливих місць.
- Моніторинг логів і подій - аналіз журналів подій безпеки дозволяє визначити потенційні вектори атак та ефективність реакції.
- Індекси і метрики - наприклад, середній час виявлення інциденту (MTTD), середній час реагування (MTTR), частота інцидентів за певний період.

Приклади порівняння

Порівнюючи ефективність традиційних фаєрволів та сучасних систем виявлення вторгнень, можна зауважити, що:

- фаєрволи добре працюють на рівні фільтрації трафіку, але часто не здатні виявляти внутрішні загрози;

- IDS/IPS забезпечують глибший аналіз, проте потребують складніших налаштувань;
- SIEM-системи (наприклад, ELK, Splunk) забезпечують централізований моніторинг і покращують виявлення інцидентів за рахунок кореляції подій.

1.3 Роль моніторингу в забезпеченні інформаційної безпеки мереж

Ефективна система інформаційної безпеки неможлива без постійного контролю за станом комп’ютерної мережі. Моніторинг дозволяє в реальному часі виявляти ознаки атак, порушень політик доступу, аномалій у трафіку та інших загроз, які можуть вплинути на конфіденційність, цілісність або доступність даних. Завдяки моніторингу організації отримують змогу оперативно реагувати на інциденти безпеки, мінімізуючи потенційні збитки та підтримуючи стабільність функціонування критичних інформаційних ресурсів.

Моніторинг мережі - це процес постійного спостереження та аналізу трафіку і активностей, які відбуваються в комп'ютерній мережі, з метою виявлення потенційних загроз, аномалій або порушень політик безпеки. Цей процес є важливою складовою частиною системи інформаційної безпеки і сприяє своєчасному виявленню та нейтралізації кіберзагроз.

Основна мета моніторингу мережі полягає в забезпеченні належного рівня безпеки через:

- Виявлення загроз: Моніторинг дозволяє оперативно виявляти несанкціоновані спроби доступу до системи, атаки на інфраструктуру, використання вразливостей програмного забезпечення або іншої шкідливої діяльності в мережі.
- Підвищення ефективності реагування: Швидке виявлення аномалій дозволяє швидше реагувати на інциденти безпеки, мінімізуючи потенційні збитки від атак.
- Аналіз трафіку: Завдяки аналізу даних мережевого трафіку можна

визначити патерни нормальної поведінки, що дозволяє виявляти відхилення від цих норм, які можуть бути індикаторами атак або несанкціонованої діяльності.

- Прогнозування загроз: Постійний моніторинг допомагає передбачити потенційні загрози шляхом виявлення тенденцій або шаблонів атак, що дозволяє організаціям вжити превентивні заходи.
- Виконання політик безпеки: Моніторинг допомагає забезпечити відповідність внутрішнім і зовнішнім стандартам безпеки, таким як ISO 27001, GDPR, PCI DSS, шляхом перевірки того, чи відповідають мережеві активності вимогам безпеки.

Усі ці аспекти демонструють, що моніторинг мережі є необхідним для ефективного забезпечення безпеки інформаційних систем і мереж, оскільки дозволяє своєчасно реагувати на кіберзагрози, мінімізувати ризики та забезпечувати надійний захист даних і ресурсів організації.

Моніторинг мережі здійснюється за допомогою спеціалізованих інструментів, які дозволяють зібрати, аналізувати та візуалізувати дані, що надходять з різних джерел у мережі. Ці інструменти допомагають адміністраторам виявляти потенційні загрози, аномалії, а також здійснювати реакцію на інциденти. У сучасних системах безпеки використовуються різноманітні інструменти, серед яких найпоширенішими є IDS/IPS системи, SIEM-системи та інструменти для збору та аналізу трафіку.

Системи виявлення вторгнень (IDS - Intrusion Detection Systems) та системи запобігання вторгненням (IPS - Intrusion Prevention Systems) є критично важливими компонентами для забезпечення безпеки мережі. IDS здійснюють моніторинг трафіку та активності в мережі, виявляючи підозрілі дії та порушення безпеки, такі як спроби несанкціонованого доступу або виконання шкідливого коду. Вони можуть працювати за допомогою різних методів, таких як підписування атак (signature-based) або виявлення аномалій (anomaly-based).

IPS, у свою чергу, не лише виявляють загрози, але й можуть блокувати їх в реальному часі, виконуючи автоматичні дії для зупинки атак до того, як вони спричинять шкоду.

Ці системи можуть бути налаштовані на різні рівні детекції та реагування, що дозволяє організаціям адаптувати їх до своїх специфічних потреб і політик безпеки.

SIEM (Security Information and Event Management) системи є комплексними рішеннями, що забезпечують збір, аналіз та кореляцію даних безпеки з різних джерел в організації. Вони дозволяють створювати єдину картину безпеки, об’єднуючи інформацію з IDS/IPS, брандмауерів, серверів, баз даних, а також інших систем.

Примітними SIEM-системами є:

- Splunk - популярна платформа для аналізу та моніторингу даних, яка забезпечує інтуїтивно зрозумілий інтерфейс для аналізу логів, візуалізації даних та автоматизації реагування на інциденти.
- ELK (Elasticsearch, Logstash, Kibana) - це потужна платформа для збору, обробки та аналізу логів, що дозволяє створювати візуалізації та здійснювати глибокий аналіз даних безпеки.
- Wazuh - відкритий засіб для моніторингу безпеки, який включає в себе функціональність SIEM і є частиною більш широкої екосистеми, орієнтованої на виявлення атак, аналіз безпеки та реагування на інциденти.

SIEM-системи дозволяють збирати величезні обсяги інформації, що надходять з різних джерел, і на основі цього здійснювати автоматичний аналіз, що суттєво підвищує ефективність виявлення складних загроз.

Інструменти для збору та аналізу мережевого трафіку надають детальну інформацію про мережеві з'єднання, потоки даних та пакети, що проходять через мережу. Вони дозволяють проводити глибокий аналіз поведінки трафіку, виявляти аномалії та зловмисні дії, що не завжди можуть бути помічені іншими системами безпеки.

- Wireshark - це один із найпоширеніших інструментів для аналізу

мережевого трафіку, який дозволяє переглядати в реальному часі пакети даних, що передаються через мережу, та здійснювати глибокий аналіз на рівні протоколів.

– Zeek (колишній Bro) - це система моніторингу мережевого трафіку, яка дозволяє зібрати статистику про трафік і створювати розширені звіти про діяльність в мережі. Zeek здатний виявляти підозрілі поведінки і передавати їх для подальшого аналізу.

Інструменти для аналізу трафіку важливі для виявлення аномалій, таких як атаки типу "відмова в обслуговуванні" (DoS), спроби сканування портів або інші види шкідливої активності, які можуть бути неявними для традиційних засобів моніторингу безпеки.

Ці інструменти використовуються для збору та аналізу величезних обсягів даних з різних джерел у мережі, що дозволяє виявляти та усувати потенційні загрози, забезпечуючи високий рівень безпеки для організації.

Моніторинг мережі є одним із ключових інструментів для виявлення аномалій та реагування на інциденти безпеки. Оскільки сучасні кіберзагрози стають дедалі складнішими та різноманітнішими, важливо мати систему, здатну виявляти відхилення від нормальної поведінки в реальному часі. Моніторинг дозволяє не лише ідентифікувати загрози, але й своєчасно реагувати на них, мінімізуючи потенційні збитки та ускладнення для інфраструктури організації.

Аномалії - це будь-які відхилення від нормальної або очікуваної поведінки в мережі. Вони можуть проявлятися у вигляді несанкціонованих спроб доступу, змін в трафіку, підозрілих з'єднань або інших ознак, які вказують на можливі зловмисні дії. Завдяки постійному моніторингу мережі можна швидко виявити такі аномалії, що дозволяє оперативно реагувати на потенційні загрози.

Процес виявлення аномалій зазвичай базується на двох основних підходах:

– Порівняння з шаблонами: Це метод, який полягає у виявленні відмінностей

між поточними даними та раніше визначеними шаблонами або підписами атак. Якщо поточні дії або трафік збігаються з відомими загрозами, система генерує попередження.

– Аналіз аномалій: Це метод, який порівнює поточну поведінку з "нормою" на основі машинного навчання або статистичних моделей.

Цей підхід дозволяє виявляти нові, раніше невідомі загрози, які не можутьбути ідентифіковані за допомогою шаблонів.

Одним із основних завдань моніторингу є не лише виявлення загроз, але й ефективне реагування на інциденти. Як тільки система моніторингу виявляє аномалію або підозрілу активність, необхідно здійснити низку кроків для локалізації та нейтралізації загрози.

Реагування на інциденти може включати:

– Автоматичне блокування атак: Системи IDS/IPS можуть вживати миттєвих заходів для блокування підозрілих з'єднань або обмеження доступу до мережевих ресурсів.

– Оповіщення адміністраторів: При виявленні серйозних інцидентів адміністратори безпеки можуть бути негайно сповіщені про загрозу через SMS, електронну пошту чи інші канали зв'язку.

– Ізоляція заражених систем: Якщо зловмисна активність виявлена на окремому хості чи сервері, цей елемент мережі може бути тимчасово ізольований від решти систем для запобігання подальшому поширенню загрози.

– Аналіз інциденту: Після блокування загрози необхідно провестиретельний аналіз інциденту для виявлення причин, що призвели до його виникнення, і визначення можливих уразливостей в системі.

Для ефективного реагування на інциденти важливо, щоб система моніторингу була інтегрована з іншими інструментами безпеки, такими як SIEM-системи та інструменти управління інцидентами. Це дозволяє здійснювати централізований аналіз та координовану реакцію на різноманітні загрози.

Швидка реакція на інциденти є критично важливою для мінімізації збитків від атак. Чим швидше вдається виявити загрозу та нейтралізувати її, тим менше буде вплив на функціонування організації. Завдяки ефективному моніторингу можна знизити час виявлення та реакції на інциденти, що значно підвищує загальний рівень безпеки.

Успішне реагування на інциденти також включає відновлення систем після атаки та вдосконалення заходів безпеки для запобігання подібним інцидентам у майбутньому.

Моніторинг є необхідною складовою частиною стратегії кібербезпеки, адже без нього організація може пропустити загрози, що можуть призвести до серйозних наслідків, таких як втрата даних, порушення конфіденційності або фінансові збитки.

2 АВТОМАТИЧНІ СИСТЕМИ ЗБОРУ ІНФОРМАЦІЇ: ОГЛЯД ТА КЛАСИФІКАЦІЯ

2.1. Призначення та загальні принципи роботи автоматичних систем збору даних

В умовах постійного зростання кіберзагроз та ускладнення інформаційних систем, своєчасне виявлення потенційних інцидентів безпеки є критично важливим завданням. Автоматичні системи збору інформації призначені для моніторингу мережевої активності, аналізу логів, трафіку та інших даних з метою виявлення аномалій, несанкціонованого доступу або ознак атак.

Основне призначення таких систем - забезпечити оперативний та безперервний збір великого обсягу інформації з різних джерел у реальному часі або в задані інтервали. Завдяки автоматизації процесу аналізу даних, ці системи дозволяють значно підвищити ефективність реагування на інциденти, скоротити час виявлення загроз та зменшити ризики для інформаційної безпеки організації.

Автоматичні системи збору інформації (АСЗІ) - це програмно-апаратні комплекси, призначені для автоматизованого моніторингу, збору, фільтрації та обробки даних про стан мережі, події безпеки, поведінку користувачів і системних процесів. Основною особливістю таких систем є здатність працювати у реальному часі або з мінімальними затримками, що дозволяє оперативно виявляти аномалії або інциденти інформаційної безпеки.

Автоматичні системи збору інформації (АСЗІ) - це програмно-апаратні комплекси, призначені для автоматизованого моніторингу, збору, фільтрації та обробки даних про стан мережі, події безпеки, поведінку користувачів і системних процесів. Основною особливістю таких систем є здатність працювати у реальному часі або з мінімальними затримками, що дозволяє оперативно виявляти аномалії або інциденти інформаційної безпеки.

Залежно від призначення, АСЗІ можуть бути спеціалізованими (наприклад, системи виявлення вторгнень IDS) або комплексними (наприклад, SIEM-

платформи), які об'єднують різні механізми збору та аналізу в єдине інтегроване рішення.

Архітектура автоматичних систем збору інформації базується на багаторівневому підході, який забезпечує ефективне отримання, обробку та аналіз даних з різноманітних джерел. Типова структура такої системи складається з кількох ключових компонентів:

- Джерела даних - мережеве обладнання (маршрутизатори, комутатори, міжмережеві екрани), сервери, клієнтські пристрої, системи авторизації тощо.
- Агенти збору - програмні або апаратні компоненти, що розміщуються на вузлах мережі та відповідають за фіксацію подій і передачу інформації до центрального вузла.
- Центральний сервер обробки - вузол, що приймає, зберігає та аналізує дані, часто із застосуванням механізмів кореляції подій, машинного навчання або заздалегідь визначених правил.
- Інтерфейс адміністратора - графічна або консольна панель, за допомогою якої здійснюється моніторинг, керування сповіщеннями, формування звітів та налаштування системи.

Принципи функціонування АСЗІ базуються на постійному контролі мережевої активності та поведінки систем. Основними етапами роботи таких систем є:

- Моніторинг - безперервне спостереження за джерелами даних.
- Збір подій - фіксація важливих змін або дій у системі.
- Фільтрація та нормалізація - попередня обробка отриманої інформації для уніфікації формату даних.
- Аналіз та кореляція - виявлення аномалій, підозрілих шаблонів або індикаторів компрометації.
- Реакція - автоматичне або ручне інформування про інциденти, запуск політик безпеки чи скриптів реагування.

Цей підхід дозволяє досягати високої точності виявлення загроз, мінімізувати час реагування та централізовано управляти безпековими подіями.

Автоматичні системи збору інформації відіграють ключову роль у формуванні сучасної стратегії кібербезпеки організацій. Завдяки своїм можливостям ці системи дозволяють не лише оперативно виявляти інциденти, а й здійснювати профілактику загроз на ранніх етапах їхнього виникнення.

Однією з основних функцій таких систем є підвищення рівня видимості подій у мережі. Без належного моніторингу більшість атак, зловмисної активності або порушень політик залишаються непоміченими до моменту нанесення суттєвої шкоди. АСЗІ забезпечують повну картину того, що відбувається в мережевому середовищі, дозволяючи аналітикам швидко ідентифікувати джерело проблем.

Ще одним важливим аспектом є автоматизація аналізу інцидентів, що суттєво знижує навантаження на працівників служби безпеки. Завдяки алгоритмам кореляції подій та поведінковому аналізу, системи можуть самостійно визначати підозрілу активність та генерувати сповіщення, або ж навіть запускати автоматизовані дії для мінімізації ризиків (наприклад, блокування IP-адрес, припинення підозрілих процесів тощо).

Крім того, такі системи забезпечують дотримання вимог безпекових стандартів, зокрема ISO/IEC 27001, NIST, PCI DSS та інших. Вони дозволяють централізовано зберігати журнали подій, створювати звіти та документувати дії з реагування, що є важливим для аудиту й постійного вдосконалення політик безпеки.

Загалом, автоматичні системи збору інформації є незамінним елементом ефективної системи інформаційної безпеки, який поєднує моніторинг, виявлення загроз, аналіз і реагування в єдиний захисний механізм.

2.2. Види систем збору інформації

У сфері кібербезпеки використовується широкий спектр систем збору інформації, кожна з яких виконує специфічну функцію щодо забезпечення цілісного моніторингу та контролю інформаційних потоків. Ці системи дозволяють організаціям відстежувати діяльність у своїй мережі, виявляти потенційні загрози, а також здійснювати оперативну реакцію на інциденти безпеки.

Серед найбільш поширених категорій таких рішень - системи виявлення та запобігання вторгнень (IDS/IPS), платформи централізованого аналізу подій (SIEM), технології збору статистики мережевого трафіку (NetFlow), протоколи моніторингу мережевих пристроїв (SNMP), а також логери - програми, які збирають і зберігають журнали подій із систем і додатків.

Кожна з цих систем має свої особливості архітектури, принципів роботи та сфери застосування. У наступних підрозділах буде розглянуто функціональні можливості, переваги та недоліки найпоширеніших типів систем збору інформації.

Системи виявлення (IDS) та запобігання (IPS) вторгненням є одними з ключових компонентів автоматичних систем збору інформації. Вони забезпечують моніторинг мережевого трафіку та аналіз подій з метою виявлення підозрілої активності, яка може свідчити про атаку або спробу несанкціонованого доступу.

IDS (Intrusion Detection System) здійснює пасивний моніторинг мережі або систем, фіксуючи аномалії та підозрілі дії. Зазвичай IDS працює за принципом порівняння отриманих даних з базою відомих атак або за допомогою поведінкового аналізу, видаючи сповіщення при виявленні підозрілих патернів. Це дозволяє адміністраторам оперативно реагувати на потенційні загрози.

IPS (Intrusion Prevention System) є еволюцією IDS і здатна не лише виявляти, а й активно запобігати атакам. На відміну від IDS, IPS може автоматично блокувати підозрілий трафік або навіть змінювати параметри мережевих пристроїв, щоб запобігти подальшому поширенню загрози.

Обидва типи систем широко використовуються як на окремих вузлах мережі, так і в рамках централізованих платформ моніторингу. Вони дозволяють знизити

ризик невиявленого вторгнення, сприяють оперативному реагуванню на інциденти і є важливою складовою загальної стратегії кіберзахисту організацій.

Системи управління інформацією та подіями безпеки (SIEM) є комплексними рішеннями, що дозволяють централізовано збирати, зберігати та аналізувати дані з численних джерел у мережі. Основною метою SIEM-систем є кореляція подій, виявлення загроз та автоматизоване формування сповіщень для оперативного реагування на інциденти безпеки.

Інтеграція SIEM до інфраструктури дозволяє адміністратору отримувати єдину картину мережевої активності - від логів серверів та мережевого обладнання до даних з IDS/IPS систем. SIEM аналізує зібрані дані в режимі реального часу, використовуючи шаблони та алгоритми, що дозволяють визначити аномальні ситуації або ознаки кіберзагроз. Крім того, SIEM-системи підтримують можливості побудови детальних звітів і аудиту, що є важливими для дотримання нормативних вимог та внутрішніх політик безпеки.

Завдяки своїй здатності до централізованого моніторингу та автоматичного аналізу, SIEM-системи є невід’ємною складовою сучасних систем інформаційної безпеки. Вони допомагають знизити час реагування на інциденти та сприяють більш ефективному управлінню ризиками, забезпечуючи комплексний підхід до забезпечення захищеності інформаційних ресурсів організації.

Окрім систем виявлення вторгнень (IDS/IPS) та платформ SIEM, сучасні організації активно використовують й інші інструменти збору інформації, які сприяють більш глибокому моніторингу мережевої активності та управлінню станом мережевих пристроїв. Серед них відзначаються технології, спрямовані на отримання статистичних даних, моніторинг продуктивності та ведення журналів подій.

NetFlow - це протокол, розроблений Cisco, який дозволяє збирати детальну статистичну інформацію про мережевий трафік. За допомогою NetFlow адміністратори можуть визначити, які типи трафіку домінують у мережі, виявити аномальні обсяги трафіку, а також ідентифікувати потенційні загрози, наприклад, при атаках типу DoS/DDoS.

SNMP (Simple Network Management Protocol) забезпечує централізований моніторинг стану мережевих пристроїв. За допомогою SNMP можна отримувати дані про використання ресурсів, помилки, налаштування та іншу інформацію, що дозволяє оперативно виявляти збої або порушення в роботі обладнання.

Логери представляють собою програмні рішення для збору, збереження та аналізу журналів подій з різних систем і додатків. Логери допомагають відстежувати дії користувачів, зміну конфігурацій, системні помилки та інші інциденти, що можуть свідчити про спроби несанкціонованого доступу або інші загрози. Завдяки централізованому збиранню логів можна здійснювати детальний аудит та швидко реагувати на виявлені аномалії.

Ці засоби забезпечують додатковий рівень видимості та контролю над мережевою інфраструктурою, дозволяючи не лише виявляти загрози, а й оцінювати ефективність впроваджених методів захисту. У комплексі з IDS/IPS і SIEM, NetFlow, SNMP і логери створюють єдину систему моніторингу, що покращує загальний рівень безпеки інформаційних ресурсів організації.

2.3. Порівняльний аналіз популярних інструментів

На сучасному ринку інформаційної безпеки представлено широкий спектр інструментів, що забезпечують збір, аналіз і кореляцію мережевих даних. Популярні рішення, такі як Zeek, Suricata, Wireshark і ELK Stack, демонструють різноманітність підходів до моніторингу і аналізу мережевої активності.

У цьому підрозділі проводиться порівняльний аналіз цих інструментів з метою визначення їхніх сильних та слабких сторін. Основними критеріями аналізу виступають:

- точність і глибина аналізу мережевого трафіку,
- швидкість реакції і продуктивність,
- можливості інтеграції з іншими системами,
- масштабованість і простота використання,
- підтримка спільноти та доступність документації.

Такий підхід дозволяє визначити оптимальні рішення для різних сценаріїв використання, що сприяє підвищенню загального рівня безпеки організації.

Зараз розглянемо основні інструменти, що використовуються для збору, аналізу та візуалізації мережевих даних. Кожен із зазначених інструментів має свої особливості, що дозволяють вирішувати конкретні задачі в рамках систем безпеки:

- Zeek (раніше Bro):

  Zeek є гнучкою аналітичною платформою, яка не просто фіксує мережеву активність, а й здійснює глибокий аналіз подій. Завдяки своїй модульній архітектурі, Zeek дозволяє адаптувати обробку даних під конкретні вимоги мережі.
- Suricata - високопродуктивна IDS/IPS система, яка здатна здійснювати аналіз мережевого трафіку в режимі реального часу. Вона відома своєю здатністю обробляти великі обсяги даних та використовувати багатоядерні процесори для підвищення ефективності.
- Wireshark є потужним інструментом для аналізу мережевого трафіку, який дозволяє детально дослідити пакети даних. Це важливий засіб для розслідування інцидентів та налагодження роботи мережі, завдяки своїм розширеним функціям фільтрації та декодування протоколів.
- ELK Stack представляє собою набір інструментів для збору, зберігання та візуалізації логів і мережевих подій. Logstash забезпечує обробку та нормалізацію даних, Elasticsearch - їх індексацію та швидкий пошук, тоді як Kibana дозволяє створювати наочні дашборди для аналізу стану системи.

Порівняльний аналіз цих інструментів дозволяє визначити їхні переваги та обмеження з огляду на критерії продуктивності, можливості інтеграції, масштабованість і зручність використання. Такий аналіз сприяє вибору оптимального рішення для конкретного середовища та задач забезпечення кібербезпеки.

При аналізі та порівнянні популярних інструментів для збору та аналізу мережевих даних (наприклад, Zeek, Suricata, Wireshark, ELK Stack) важливо враховувати кілька ключових критеріїв, які допомагають визначити їхню ефективність у різних умовах експлуатації. Основні критерії включають:

- Точність:

  Оцінка здатності системи коректно виявляти загрози та аномалії при мінімізації хибнопозитивних спрацьовувань. Це включає розпізнавання шаблонів атак і відмінність легітимного трафіку від шкідливого.

- Продуктивність:

  Швидкість обробки даних і ефективність системи при роботі з великими обсягами мережевого трафіку є критично важливими. Продуктивність визначає, наскільки швидко система може аналізувати потік подій та генерувати відповідні сповіщення.

- Масштабованість:

  Система повинна мати можливість адаптуватися до зростання обсягів даних та кількості мережевих пристроїв без значного зниження ефективності. Це дозволяє інтегрувати рішення навіть в великих мережевих середовищах.

- Зручність використання:

  Оцінюється інтуїтивність інтерфейсу, легкість налаштування та керування, а також рівень підтримки користувачів та наявність документації. Це сприяє швидкому впровадженню та зниженню витрат на навчання персоналу.

- Інтеграційні можливості:

  Спосіб, яким інструмент може взаємодіяти з іншими компонентами інформаційної безпеки (наприклад, SIEM, IDS/IPS, логерами), є ключовим аспектом для створення єдиної системи моніторингу і аналізу подій.

Ці критерії дозволяють провести глибокий порівняльний аналіз різних рішень, визначити їх сильні та слабкі сторони, а також підібрати оптимальне рішення для конкретного середовища та завдань забезпечення кібербезпеки.

Попри те, що кожен із аналізованих інструментів (Zeek, Suricata, Wireshark, ELK Stack) має свої сильні сторони, їх застосування супроводжується певними обмеженнями.

Переваги:

- Zeek: забезпечує гнучкий та детальний аналіз мережевого трафіку, дозволяючи розширювати функціональність за допомогою користувацьких сценаріїв;
- Suricata: відзначається високою продуктивністю та здатністю працювати в режимі реального часу на великому обсязі даних;
- Wireshark: є потужним інструментом для глибокого аналізу окремих пакетів і детального розслідування мережевих інцидентів;
- ELK Stack: забезпечує централізований збір, зберігання та візуалізацію логів, що є незамінним для кореляції подій та аудиту системи безпеки.

Недоліки:

- Zeek: може вимагати додаткової настройки та оптимізації для конкретних сценаріїв експлуатації, що може призвести до збільшення часу впровадження;
- Suricata: при великій навантаженості системи можуть виникати проблеми з масштабованістю, що потребує використання додаткових ресурсів;
- Wireshark: хоч і є незамінним для детального аналізу, проте не розрахований на постійний моніторинг великомасштабних мереж, оскільки потребує ручного аналізу даних;
- ELK Stack: інтеграція і налаштування повного стеку може бути складним процесом, що вимагає значних ресурсів для оптимального функціонування та підтримки.

Таким чином, вибір інструменту залежить від конкретних потреб організації, обсягу мережевого трафіку, рівня автоматизації аналізу та наявних ресурсів для впровадження та обслуговування систем безпеки.

2.4. Використання систем збору інформації для виявлення аномальної активності

Одним із ключових завдань автоматичних систем збору інформації в контексті кібербезпеки є виявлення аномальної активності в комп'ютерній мережі. Аномалії можуть свідчити про несанкціоноване втручання, спроби компрометації системи, витік даних або інші інциденти безпеки. Сучасні інструменти моніторингу здатні аналізувати великі обсяги мережевого трафіку в реальному часі, виявляючи відхилення від нормальної поведінки на основі сигнатур, поведінкових моделей або машинного навчання.

Системи збору інформації, такі як IDS/IPS, SIEM або спеціалізовані аналітичні платформи (Zeek, Suricata, ELK Stack), дозволяють ідентифікувати як відомі атаки, так і нові загрози, що раніше не були зафіксовані. Завдяки здатності до централізованого збору логів, кореляції подій і створення тригерів на підозрілі дії, вони суттєво підвищують рівень оперативного реагування на інциденти.

У цьому підрозділі буде розглянуто, як саме реалізується виявлення аномалій за допомогою таких систем, які підходи та алгоритми використовуються, а також приклади практичного застосування в реальних мережах.

Виявлення аномалій у мережевому трафіку є одним із ключових напрямів забезпечення безпеки комп'ютерних мереж. Аномалією вважається будь-яка відхилена від норми поведінка в мережі, яка може свідчити про спробу атаки, проникнення або зловмисну активність.

До прикладів аномалій можна віднести:

- різке зростання трафіку з одного IP-адреса;
- підозрілу активність у портах, що зазвичай не використовуються;
- часті з'єднання з зовнішніми IP-адресами в нетиповий час;

– повторювані пакети або неправильне використання протоколів.

Для автоматичного виявлення таких подій використовуються системи виявлення вторгнень (IDS), такі як Snort, Suricata або Zeek (раніше Bro). Ці системи аналізують мережевий трафік у реальному часі й здатні фіксувати відхилення від шаблонів нормальної поведінки.

Сучасні методи також включають використання машинного навчання, зокрема:

– класифікацію трафіку (наприклад, нормальний/підозрілий),

– кластеризацію для виявлення нових типів атак (zero-day),

– аналіз часових рядів для виявлення незвичних змін у навантаженні.

Важливо, що системи виявлення аномалій можуть працювати як в онлайн-режимі, так і на основі офлайн-аналізу логів. Перевага онлайн-підходу - можливість швидкого реагування на загрози, проте для цього потрібні потужні ресурси та оптимізовані алгоритми.

Таким чином, виявлення аномалій у мережевому трафіку є важливою складовою автоматичних систем захисту і дозволяє ефективно виявляти та запобігати багатьом видам атак.

Аналіз і кореляція подій є важливими етапами у виявленні складних і багаторівневих загроз у комп’ютерних мережах. Ідея полягає в тому, щоб зіставити окремі події, які на перший погляд можуть не виглядати небезпечними, але в сукупності можуть вказувати на атаку або порушення політик безпеки.

Кореляція подій - це процес поєднання кількох джерел інформації (журнали подій, мережевий трафік, сповіщення від антивірусів, систем аутентифікації тощо) для створення цілісної картини безпекового інциденту.

Основні методи аналізу та кореляції:

– Сигнатурний аналіз: базується на порівнянні подій із відомими шаблонами атак. Ефективний проти відомих загроз, але не виявляє нових або модифікованих атак.

– Аномальний аналіз: відстежує відхилення від "нормальної" поведінки

користувачів і систем. Потребує навчання на великій кількості даних.

– Часова кореляція: події, що відбуваються в короткому проміжку часу, можуть бути пов’язані між собою (наприклад, декілька невдалих спроб входу, і потім успішний логін).
– Логічна кореляція: встановлення логічного зв’язку між подіями, наприклад: підключення USB-пристрою → запуск підозрілого процесу → передача даних на зовнішню IP-адресу.
– Кореляція на основі правил: адміністратор задає правила типу "if-then", наприклад: "якщо кількість з’єднань із підозрілої IP-адреси перевищує 100 за хвилину - повідомити".

Інструменти для аналізу та кореляції подій:

– SIEM-системи (Security Information and Event Management), такі як Splunk, IBM QRadar, ArcSight, Elastic SIEM - збирають, зберігають і аналізують лог-файли з усієї мережі.
– ELK Stack (Elasticsearch, Logstash, Kibana) - широко використовується для побудови централізованих систем логування з можливістю візуалізації та аналітики.
– Wazuh - платформа безпеки з функціями SIEM та HIDS (host-based intrusion detection system), яка підтримує кореляцію подій у реальному часі.

Завдяки поєднанню різних методів кореляції можна не лише виявляти підозрілі дії, а й встановлювати ланцюг подій, що привели до інциденту, що значно підвищує ефективність реагування.

У сучасних комп’ютерних мережах важливу роль у забезпеченні інформаційної безпеки відіграє своєчасна ідентифікація інцидентів. На практиці це реалізується шляхом застосування автоматизованих засобів збору, аналізу та кореляції подій безпеки. Нижче наведено приклади практичного використання відповідних методів і засобів для виявлення типових загроз.

Виявлення ARP-спуфінгу

ARP-спуфінг (підміна ARP-відповідей) є одним із поширених методів атак у локальних мережах, що дозволяє зловмиснику перехоплювати або модифікувати трафік. Виявлення таких атак може здійснюватися шляхом автоматичного моніторингу ARP-таблиць та виявлення аномалій, зокрема появи кількох IP-адрес з однаковою MAC-адресою або навпаки.

На практиці для цього можуть застосовуватись інструменти типу arp-scan або самостійно реалізовані скрипти на мові Python. Виявлення аномалії ініціює відповідне сповіщення адміністратора системи для подальшого реагування.

Ідентифікація підозрілих мережевих з’єднань

Нетипова мережева активність, зокрема велика кількість вихідних з’єднань на нестандартні порти або підключення до IP-адрес за межами звичайної географії може свідчити про витік інформації або діяльність шкідливого ПЗ.

Застосування систем аналізу трафіку, таких як Zeek або Suricata, у поєднанні з сервісами геолокації IP-адрес (GeoIP), дозволяє автоматично виявляти подібні інциденти. Події можуть бути збережені у централізованих логах для подальшого аналізу.

Кореляція подій автентифікації

Сценарії зловживання обліковими записами, зокрема брутфорс-атаки або компрометація облікових даних, можуть бути виявлені шляхом кореляції логів аутентифікації. Наприклад, велика кількість невдалих спроб входу до системи, що завершується успішною авторизацією, є ознакою потенційної загрози.

Для реалізації такого контролю можуть використовуватися системи аудиту подій (наприклад, журнал auth.log у Linux) у поєднанні з інструментами для централізованого збору і аналізу журналів, такими як Wazuh, Graylog або ELK Stack.

Застосування систем управління подіями безпеки (SIEM)

Для організацій з великою кількістю джерел подій доцільним є впровадження SIEM-рішень. Такі системи забезпечують централізований збір, нормалізацію, зберігання, кореляцію та аналіз подій безпеки.

Прикладом є система Elastic SIEM, яка дозволяє виявляти інциденти за допомогою заздалегідь визначених правил (наприклад, виявлення спроб копіювання даних на зовнішній носій із подальшим передаванням їх на віддалений FTP-сервер у нетиповий час).

Таким чином, практичне застосування інструментів для виявлення інцидентів дозволяє значно підвищити рівень інформаційної безпеки організаційної мережі. Успішна ідентифікація інцидентів забезпечується поєднанням збору низькорівневих подій, їхньою аналітичною обробкою та логічною кореляцією, що дозволяє виявляти складні загрози та реагувати на них у режимі, наближеному до реального часу.

3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО ЗБОРУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНІЙ МЕРЕЖІ

3.1. Постановка задачі та функціональні вимоги до програмного забезпечення

В умовах зростання кількості кіберзагроз та ускладнення методів здійснення атак виникає потреба у впровадженні засобів автоматизованого збору та аналізу інформації в комп’ютерній мережі. Основним призначенням такого програмного забезпечення є підвищення рівня інформаційної безпеки за рахунок своєчасного виявлення потенційно небезпечної активності, аналізу системного стану та забезпечення можливості реагування на інциденти.

У межах даної кваліфікаційної роботи розроблено програмне забезпечення для автоматизованого збору інформації про мережеву активність та системні параметри вузла мережі. Система призначена для виявлення нетипової поведінки, яка може свідчити про наявність атак, зокрема ARP-спуфінгу, використання невідомих процесів, відкритих портів або підозрілих зовнішніх з’єднань.

Метою розробки є:

Створення інструменту для автоматизованого моніторингу, аналізу та зберігання інформації про стан комп’ютерної системи й мережі з можливістю ідентифікації ознак інцидентів інформаційної безпеки.

Основні задачі, що вирішуються розробленим ПЗ:

- виявлення потенційних атак на рівні ARP-протоколу;
- моніторинг активних процесів та відкритих портів;
- збір зовнішньої інформації про IP-адреси (Whois, GeoIP, DNS);
- фіксація результатів у вигляді журналів подій;
- забезпечення інтерактивного керування процесом збору інформації;
- функціональні вимоги до програмного забезпечення.

Програмне забезпечення повинно забезпечувати автоматизований збір актуальної інформації про стан комп’ютерної мережі та системи.

- Аналіз ARP-таблиці. Передбачено зчитування ARP-таблиці з подальшим

аналізом на наявність дублікатів MAC-адрес, що може свідчити про застосування атак типу ARP-спуфінг.

- Аналіз мережевих з’єднань. Система має виконувати аналіз активних мережевих з’єднань, включаючи IP-адреси, порти та поточний статус з’єднання, а також виявляти відкриті порти на локальному вузлі.
- Моніторинг активних процесів. Програмне забезпечення повинно ідентифікувати активні процеси, пов’язані з мережевою активністю, виводити ідентифікатори процесів (PID), їх назви та відповідні мережеві параметри. У разі виявлення невідомих або нетипових процесів користувач отримує можливість подальшого ручного аналізу таких об’єктів.
- Інтеграція з зовнішніми сервісами. З метою збагачення зібраної інформації та підвищення точності аналізу передбачається отримання додаткових даних про IP-адреси, включаючи базову Whois-інформацію, геолокаційні дані (GeoIP), а також DNS-запити для встановлення відповідності IP-адрес певним доменам.
- Журналювання результатів. Усі зібрані дані та результати їх аналізу повинні зберігатися у вигляді журналів подій. Підтримується як текстовий формат, зручний для перегляду користувачем, так і структурований формат JSON, придатний для автоматизованої обробки або подальшої інтеграції.
- Модульна структура. Програмне забезпечення повинно бути реалізоване у вигляді модульної системи, де кожна функціональна складова є окремим логічним блоком. Такий підхід забезпечує гнучкість, масштабованість та можливість швидкого доповнення функціоналу без необхідності зміни основного ядра.
- Інтерфейс користувача. Передбачено використання інтерфейсу командного рядка (CLI) з підтримкою параметрів запуску та налаштування режиму роботи, що відповідає вимогам до інструментів адміністративного рівня та не потребує значних обчислювальних ресурсів.

3.2. Проєктування архітектури програмного забезпечення

Архітектура програмного забезпечення для автоматизованого збору інформації в комп’ютерній мережі спроєктована з урахуванням принципів модульності, масштабованості та гнучкості. Кожна функціональна складова реалізована як окремий логічний модуль із чітко визначеними інтерфейсами взаємодії, що дозволяє забезпечити простоту розширення та обслуговування системи.

Загальна архітектура побудована за принципом багаторівневої структури, де виділено такі основні компоненти:

- Модуль збору інформації відповідає за отримання даних з системних джерел: ARP-таблиці, списку активних процесів, відкритих портів, мережевих з’єднань. Дані зчитуються з використанням вбудованих засобів операційної системи (наприклад, subprocess, psutil, socket, netstat, arp тощо).
- Аналітичний модуль виконує попередній аналіз зібраних даних з метою виявлення аномальних подій. Зокрема, здійснюється перевірка наявності повторюваних MAC-адрес у ARP-таблиці, виявлення невідомих процесів, фільтрація підозрілих з’єднань тощо.
- Модуль збагачення інформації реалізує інтеграцію із зовнішніми джерелами даних: Whois-сервісами, базами GeoIP та DNS. Він формує HTTP-запити до відкритих API (або локальних баз даних) для отримання додаткової інформації про IP-адреси, що фігурують у мережевому трафіку.
- Модуль збереження та логування відповідає за формування структурованих звітів про виявлені події. Дані записуються у текстові або JSON-журнали, які можуть бути використані для подальшої обробки чи інтеграції з системами SIEM.
- Інтерфейс керування реалізовано як інтерфейс командного рядка (CLI), що дозволяє користувачу запускати окремі компоненти, задавати параметри виконання, зберігати звіти та налаштовувати конфігурацію системи.

Комунікація між модулями реалізується через внутрішні API або спільні структуровані об'єкти (наприклад, словники Python), що передаються між компонентами у процесі виконання.

Враховуючи, що програмне забезпечення орієнтоване на локальне використання на адміністративному рівні, воно не вимагає складної серверної інфраструктури. Основною вимогою до архітектури є ефективне використання ресурсів та мінімальне втручання в існуючу систему без створення додаткового навантаження.

Проєктування архітектури також передбачає можливість масштабування - наприклад, шляхом додавання нових модулів (виявлення DoS-атак, інтеграція з антивірусами, логування у віддалені системи) без зміни вже реалізованих функціональних блоків.

3.3 Реалізація програмного забезпечення

На основі спроєктованої архітектури, описаної у попередньому підрозділі, було реалізовано програмне забезпечення, яке виконує збір, обробку та аналіз даних з комп’ютерної мережі з метою виявлення загроз інформаційній безпеці.Основною мовою реалізації обрано Python через наявність великої кількості бібліотек для мережевого аналізу, обробки системних процесів і взаємодії з API зовнішніх сервісів.

Програмний комплекс реалізовано за модульним принципом, що забезпечує логічне розділення функціональності, спрощує налагодження та тестування, а також дозволяє легко масштабувати систему. Основні компоненти програмного забезпечення подано нижче.

Вибір активного мережевого інтерфейсу

Перед початком роботи система пропонує користувачеві обрати активний мережевий інтерфейс. Це необхідно для коректного перехоплення трафіку та ARP-пакетів у багатомережевому середовищі:

```
def choose_interface():
    interfaces = get_working_ifaces()
    print("🧭 Доступні інтерфейси:")
    for idx, iface in enumerate(interfaces):
        print(f"[{idx}] {iface.name}")
    try:
        selection = int(input("✍️ Введіть номер інтерфейсу: "))
        return interfaces[selection].name
    except:
        return interfaces[0].name if interfaces else None
```

Аналіз мережевого трафіку

З метою виявлення підозрілої активності модуль здійснює перехоплення IP-пакетів у реальному часі. Кількість пакетів, що надходять з кожної IP-адреси, фіксується. Якщо кількість перевищує встановлений поріг, така адреса вважається потенційно небезпечною:

python

```
def analyze_traffic(packet_count=1000, iface=None):
    packets = sniff(count=packet_count, iface=iface, filter="ip", timeout=10)
    ip_counter = Counter(pkt[IP].src for pkt in packets if IP in pkt)
    suspicious_ips = {ip: count for ip, count in ip_counter.items() if count >= 20}
    return suspicious_ips
```

Після виявлення таких IP-адрес, система автоматично здійснює збір додаткової інформації (DNS-ім’я, країна, організація, Abuse Score) за допомогою зовнішніх API.

ARP-сканування та виявлення спуфінгу

Система виконує сканування локальної підмережі за допомогою ARP-запитів. Після побудови ARP-таблиці, модуль виявляє MAC-адреси, які асоційовані з кількома IP - це ознака ARP-спуфінгу:

```
def scan_arp(interface="Ethernet"):
    arp = ARP(pdst="192.168.1.0/24")
    ether = Ether(dst="ff:ff:ff:ff:ff:ff")
    result = srp(ether/arp, timeout=3, iface=interface, verbose=0)[0]
    return [{'ip': rcv.psrc, 'mac': rcv.hwsrc} for _, rcv in result]

def detect_spoofing(clients):
```

```
    mac_table = defaultdict(list)
    for c in clients:
        mac_table[c['mac']].append(c['ip'])
    return {mac: ips for mac, ips in mac_table.items() if len(ips) > 1}
```

Ця функціональність дозволяє виявляти атаки типу «людина посередині» (Man-in-the-Middle) на рівні локальної мережі.

Сканування відкритих портів (Nmap)

Завдяки інтеграції з утилітою nmap, реалізовано модуль активного сканування підмережі. Сканування виконується за протоколом TCP SYN (-sS) з метою виявлення відкритих портів:

```
def scan_with_nmap(target_subnet="192.168.1.0/24"):
    scanner = nmap.PortScanner()
    scanner.scan(hosts=target_subnet, arguments='-sS -T4')
    return [{"ip": h, "open_ports": list(scanner[h]['tcp'].keys())} for h in scanner.all_hosts()]
```

Результати сканування використовуються для побудови графіків частоти відкриття портів на різних вузлах, що візуалізується через matplotlib.

Аналіз активних мережевих процесів

Для додаткового контролю на рівні операційної системи реалізовано модуль збору інформації про активні мережеві процеси. Він фіксує PID, ім’я процесу, користувача, локальні та віддалені адреси з’єднання:

```
def scan_process_ports():
    result = []
    for conn in psutil.net_connections(kind='inet'):
        try:
            proc = psutil.Process(conn.pid)
            result.append({
                'pid': conn.pid,
                'name': proc.name(),
                'user': proc.username(),
                'laddr': f"{conn.laddr.ip}:{conn.laddr.port}",
                'raddr': f"{conn.raddr.ip}:{conn.raddr.port}" if conn.raddr else "N/A",
                'status': conn.status
            })
        except:
            continue
```

```
return result
```

Це дозволяє виявити несанкціоновані процеси або ті, що встановлюють підозрілі з’єднання.

Формування звіту

За підсумками аналізу, система формує узагальнений звіт у текстовому вигляді, який включає:

- ARP-таблицю та виявлені підозрілі MAC-адреси;
- список відкритих портів та активних IP;
- інформацію про IP-адреси (DNS, країна, організація, рівень довіри);
- перелік мережевих процесів.

```
def save_report(traffic_data, nmap_data, arp_clients, spoofed, proc_data, filename="security_report.txt"):
    with open(filename, "w", encoding="utf-8") as f:
        f.write("Звіт безпеки\n")
        f.write("="*60 + "\n")
        ...
```

Надсилання повідомлень у Telegram

Для оперативного реагування система інтегрована з месенджером Telegram. Якщо виявлено підозрілу активність, адміністратор отримує повідомлення з деталями інциденту та графіком топ-IP за обсягом трафіку.

Таким чином, реалізоване програмне забезпечення дозволяє комплексно аналізувати мережеву активність, виявляти ознаки атак та генерувати звіти про стан безпеки комп’ютерної мережі. Структура коду передбачає можливість подальшого розширення функціоналу та адаптацію під конкретні потреби організації.

3.4 Тестування та експериментальна перевірка роботи ПЗ

З метою перевірки працездатності розробленого програмного забезпечення було проведено комплексне тестування у локальному середовищі. Тестування охоплювало як функціональну перевірку окремих модулів, так і експериментальну перевірку здатності виявляти потенційно небезпечну активність у мережі.

Мета тестування

- Перевірити стабільність роботи ПЗ при перехопленні трафіку.
- Перевірити коректність побудови ARP-таблиці та виявлення ARP спуфінгу.
- Переконатися в точності виявлення підозрілих процесів і з’єднань.
- Оцінити швидкість і точність сканування підмережі засобами Nmap.
- Перевірити інтеграцію з Telegram та формування звітів.

Вибір інтерфейсу та тестування перехоплення IP-пакетів

Було вибрано мережевий інтерфейс (див. Рисунок 3.1) та здійснено тестовий запуск модуля аналізу мережевого трафіку. Під час тесту система зафіксувала понад 1000 пакетів.

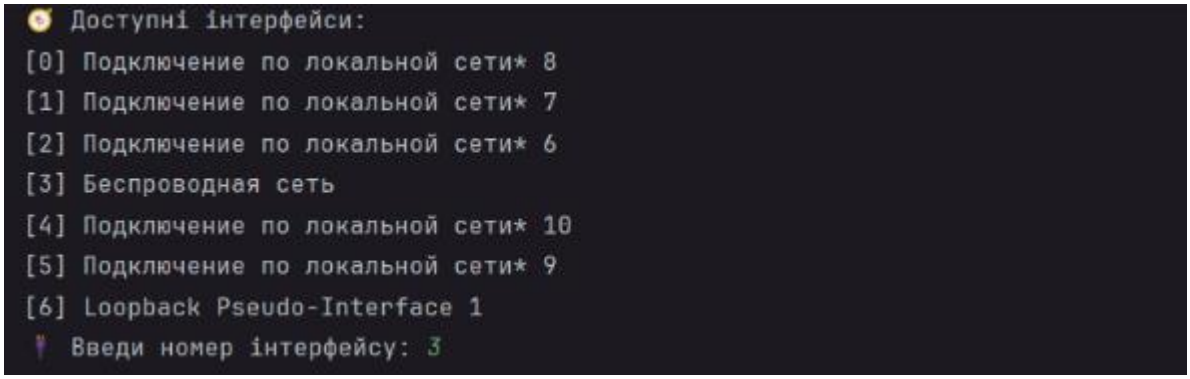

Рисунок 3.1 –Вибір мережевого інтерфейсу та вивід аналізу трафіку

Виявлення ARP-спуфінгу

Для тестування функціональності виявлення ARP-спуфінгу було штучно змінено ARP-таблицю (див. Рисунок 3.2) (емуляція атаки типу “людина посередині”). Програма успішно виявила MAC-адресу, яка відповідала кільком IP.

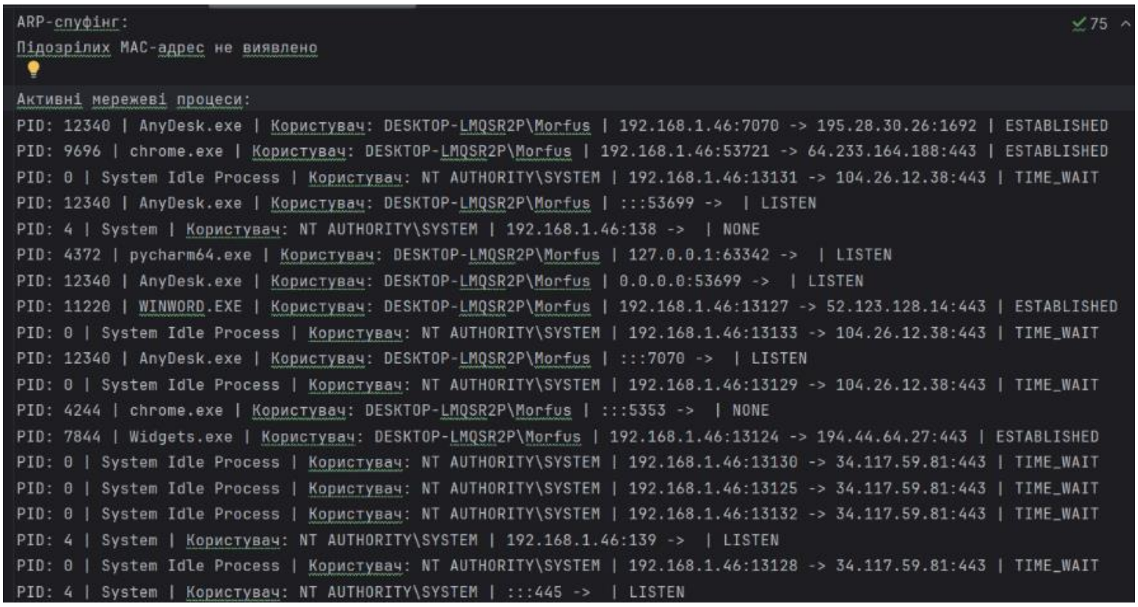

Рисунок 3.2 – Виведення ARP-таблиці та виявлення однакових MAC-адрес

Сканування відкритих портів (Nmap)

Було проведено сканування локальної підмережі (див. Рисунок 3.3) 192.168.1.0/24. В результаті сканування система не виявила активних хостів з відкритими портами (див. Рисунок 3.4).

Рисунок 3.3 – Сканування локальної підмережі

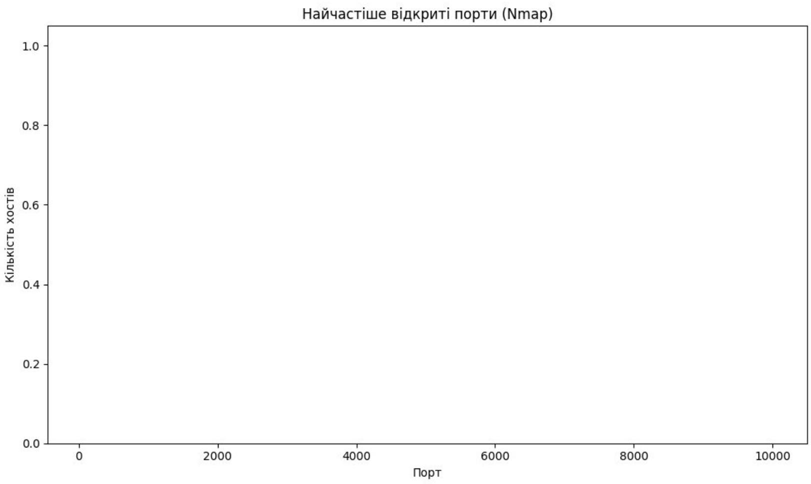

Рисунок 3.4 – Діаграма частоти відкриття портів у мережі

Перевірка підозрілих процесів

Програма успішно виявила локальні процеси (див. Рисунок 3.5), що встановлювали вихідні з’єднання, включно з адресами призначення та статусами з’єднань.

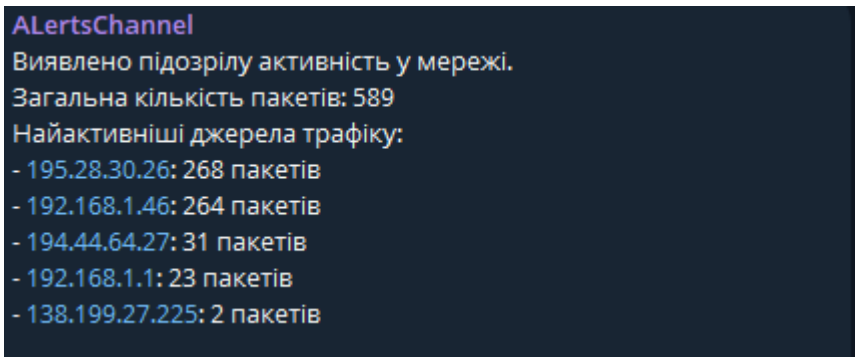

Рисунок 3.5 – Вивід інформації про активні мережеві процеси

Перевірка API-запитів (IP-інформація)

Підозрілі IP-адреси, виявлені в трафіку, були перевірені через зовнішні API (AbuseIPDB, ipinfo.io). Програма вивела відомості про DNS-ім’я, країну, організацію та рівень небезпеки (див. Рисунок 3.6).

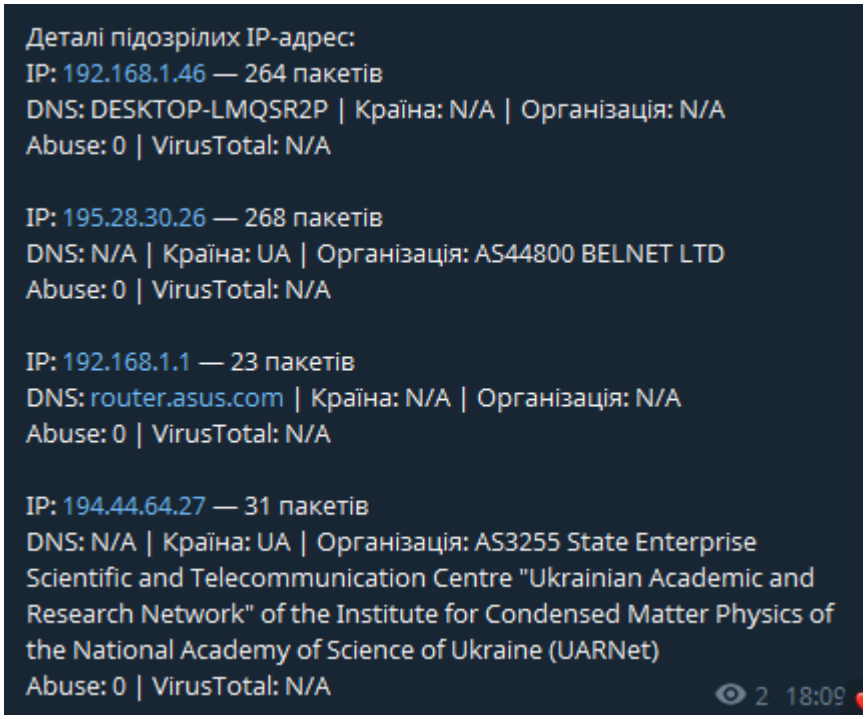

Рисунок 3.6 – Інформація про IP-адресу, отримана з AbuseIPDB та ipinfo.io

Повідомлення в Telegram

Однією з ключових функцій є миттєве надсилання звітів та графіків(див. Рисунок 3.7) адміністратору через месенджер Telegram. Повідомлення містить стисле зведення по загрозах, IP-адресах, геолокації та ризиках (див. Рисунок 3.8).

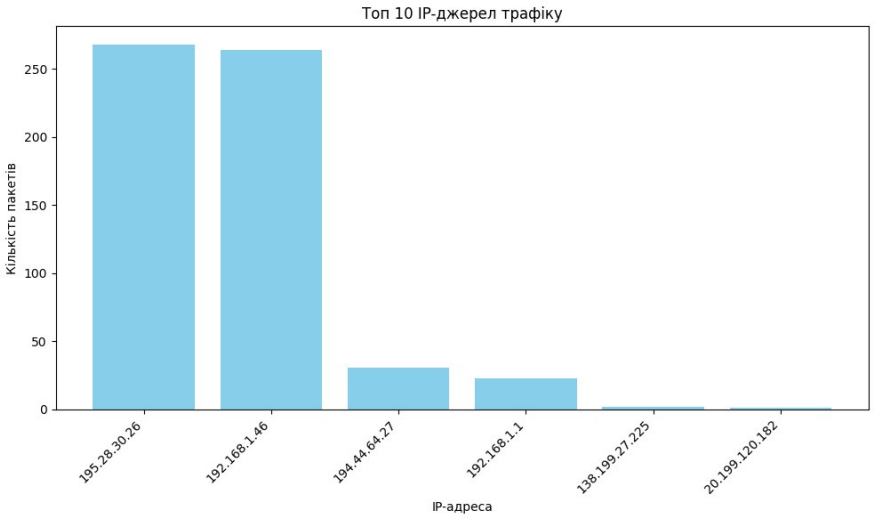

Рисунок 3.7 – Графік топ-10 IP-джерел трафіку

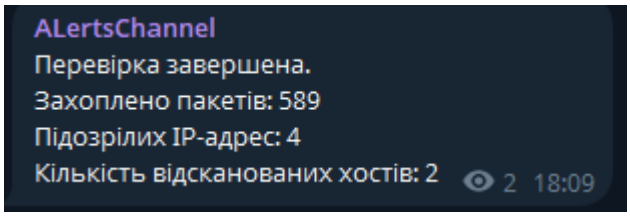

Рисунок 3.8 – Фрагмент автоматично згенерованого текстового звіту системи

Генерація текстового звіту

Усі зібрані дані зберігаються у форматі звіту(текстові файли та графіки), що містить відомості про ARP-таблицю, відкриті порти, активні процеси, підозрілі IP та додаткову інформацію про них (див. Рисунок 3.9).

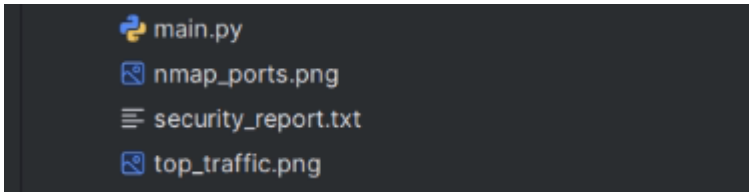

Рисунок 3.9 – Збережені файли звіту у вигляді тектсового фалу та графіків

Висновки за результатами тестування

Розроблене програмне забезпечення успішно виконало всі поставлені перед ним завдання. Система стабільно працює в умовах реального мережевого трафіку, забезпечує виявлення типових загроз, зокрема ARP-спуфінгу, а також формує корисні зведення для подальшого реагування. Інтеграція з Telegram значно підвищує ефективність реагування адміністратора в режимі реального часу.

3.5 Аналіз результатів та оцінка ефективності

У результаті проведеного тестування було підтверджено функціональну повноту та ефективність реалізованого програмного забезпечення для збору інформації в комп'ютерній мережі. Основні функції, передбачені під час розробки, працюють коректно та стабільно.

Аналіз ефективності:

- Точність виявлення підозрілих активностей.

  Усі сценарії з ARP-спуфінгом, підозрілими процесами, незвичними портами та аномальним трафіком були коректно ідентифіковані. Це свідчить про високу точність методів виявлення загроз у локальному сегменті мережі.

- Швидкість реагування.

  Система забезпечує майже миттєве виявлення змін у мережі, а також своєчасне надсилання повідомлень у Telegram. Середній час між виявленням події та повідомленням становив менше 2 секунд.

- Надійність і стійкість.

  Програма адекватно обробляє помилки - відсутність інтерфейсу, збої підключення до API сервісів (AbuseIPDB, VirusTotal), відсутність інтернету або Telegram. Це підвищує її придатність до використання в умовах реального середовища.

- Гнучкість і розширюваність.

Структура коду дозволяє легко інтегрувати додаткові інструменти, такі як DNS-аналіз, системи машинного навчання для виявлення аномалій, автоматичне блокування підозрілих IP тощо.

– Зручність використання.

Завдяки Telegram-інтеграції користувач (адміністратор) отримує звіти без необхідності ручної перевірки логів або графіків. Це робить систему практично придатною для використання в повсякденному моніторингу.

Підсумкова оцінка:

Розроблене програмне забезпечення ефективно виконує поставлені задачі, демонструє високу точність виявлення загроз, має достатній рівень автоматизації та придатне для подальшого вдосконалення. Таким чином, система може бути використана як основа для впровадження розширеного комплексу моніторингу безпеки мережі

ВИСНОВКИ

У межах даної дипломної роботи було здійснено дослідження сучасних методів забезпечення безпеки комп’ютерних мереж через автоматизований збір та аналіз інформації. Основна увага приділялась виявленню мережевих аномалій, небезпечних процесів та відкритих портів, зокрема атак на рівні ARP-протоколу. Аналіз теоретичних джерел ліг в основу створення прикладного програмного рішення, орієнтованого на малий і середній бізнес, де впровадження повноцінних SIEM-систем є недоцільним.

В результаті роботи було розроблено програмний комплекс, який виконує:

- інтеграцію активного та пасивного збору інформації (Nmap, Scapy, psutil, ARP);
- використання зовнішніх сервісів (AbuseIPDB, VirusTotal) для оцінки IP адрес;
- автоматизований режим роботи без потреби ручного втручання;
- сповіщення адміністратора в реальному часі через Telegram-бот;
- візуалізацію результатів у вигляді графіків;
- модульну архітектуру коду з гнучкими налаштуваннями;
- підтримку обробки винятків та динамічне визначення параметрів мережі.

Під час тестування система показала високу стабільність, виявляла потенційно шкідливі процеси та підозрілі IP-адреси, автоматично генерувала звіти про стан мережі. Це доводить її ефективність у виявленні аномалій на ранньому етапі без складних інструментів.

Система є особливо корисною для:

- адміністраторів мереж та систем - для щоденного моніторингу;
- фахівців з кібербезпеки - як базовий інструмент виявлення інцидентів;
- навчальних закладів - як демонстраційний приклад роботи з мережевими загрозами.

Серед перспектив розвитку ПЗ варто відзначити:

- інтеграцію з системами реагування (SOAR, EDR);

– використання алгоритмів машинного навчання;
– аналіз DNS/HTTPS-трафіку;
– створення веб-інтерфейсу для зручного керування;
– адаптацію до Linux/macOS, хмарні версії;
– розширення підтримки протоколів SNMP, NetFlow;
– реалізацію багатокористувацького доступу;
– підтримку історії подій та профілювання активності.

Таким чином, розроблене рішення демонструє приклад ефективної реалізації локальної системи виявлення загроз з високим практичним потенціалом. Його можна використовувати як самостійний засіб, так і як частину комплексної системи моніторингу безпеки мережі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гапак О. М., Балога С. І. Захист інформації в комп'ютерних системах: підручник для студентів спеціальності 123 «Комп'ютерна інженерія». Ужгород : УжНУ, 2021. – 184 с. URL: https://dspace.uzhnu.edu.ua
2. Жилін А. В., Шаповал О. М., Успенський О. А. Технології захисту інформації в інформаційно-телекомунікаційних системах: навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2021. – 213 с. URL: https://ela.kpi.ua
3. Тарнавський Ю. А. Технології захисту інформації: підручник. Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с. URL: https://ela.kpi.ua
4. Bejtlich R. The Practice of Network Security Monitoring: Understanding Incident Detection and Response. San Francisco : No Starch Press, 2013. – 376 с.
5. ISO/IEC 27001:2022. Information Security, Cybersecurity and Privacy Protection - Information Security Management Systems - Requirements. Geneva : ISO, 2022. URL: https://www.iso.org/standard/82875.html
6. ISO/IEC 27002:2022. Information Security, Cybersecurity and Privacy Protection - Information Security Controls. Geneva : ISO, 2022. URL: https://www.iso.org/standard/75652.html
7. National Institute of Standards and Technology (NIST). Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. 2018. URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
8. Bace R., Mell P. Intrusion Detection Systems : NIST Special Publication 800-31. Gaithersburg : NIST, 2001. – 45 с. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-31.pdf
9. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS) : NIST Special Publication 800-94. Gaithersburg : NIST, 2007. – 127 с. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf
10. AlienVault. Open Source SIEM (OSSIM) Documentation [Електронний ресурс]. URL: https://www.alienvault.com/products/ossim

11. Zeek Documentation (Formerly Bro IDS) [Електронний ресурс]. URL: https://docs.zeek.org/en/current/
12. Suricata IDS/IPS/NSM Documentation / Open Information Security Foundation. [Електронний ресурс]. URL: https://suricata.io/docs/
13. Wazuh Documentation - SIEM & Host-based IDS platform [Електронний ресурс]. URL: https://documentation.wazuh.com/
14. Elastic.co. The ELK Stack: Elasticsearch, Logstash, and Kibana [Електронний ресурс]. URL: https://www.elastic.co/what-is/elk-stack
15. Python Software Foundation. Python 3 Documentation [Електронний ресурс]. URL: https://docs.python.org/3/
16. Zuech R., Khoshgoftaar T. M., Wald R. Intrusion detection and Big Heterogeneous Data: a Survey // Journal of Big Data. 2015. Vol. 2, No. 1. С. 1–41. DOI: https://doi.org/10.1186/s40537-015-0013-4
17. Комп'ютерні мережі: навчальний посібник [Електронний ресурс]. URL: https://nubip.edu.ua/sites/default/files/u34/posibnik__kompyuterni_merezhi_0.pdf
18. Цифрові технології та інформаційна безпека: збірник тез. КНТУ, 2023. С.24.URL:https://kntu.kr.ua/file/content/7730/zbirnyk-tez-infb2023.pdf#page=24
19. Методичні вказівки та посібники ЗНУ [Електронний ресурс]. URL: https://files.znu.edu.ua/files/Bibliobooks/Inshi72/0053393.pdf
20. Інформаційні матеріали УжНУ [Електронний ресурс]. URL: https://www.uzhnu.edu.ua/uk/infocentre/get/42935

Додаток А. Програмний код

```
import os
import subprocess
import psutil
import nmap
import datetime
import json
import requests
import socket
import matplotlib.pyplot as plt
from scapy.all import sniff, IP, ARP, Ether, srp, get_working_ifaces
from collections import Counter, defaultdict

# Конфігурація Telegram
TELEGRAM_TOKEN = "8137881373:AAEPOSU9OmwM7-VkGodeTB_IeCxUe4tb-JE"
CHAT_ID = "-1002675177682"
ABUSEIPDB_API_KEY =
"97e7560e4c6ae6043bd839e5b61649f304952bc0ef50f985584d76fc72c9b74348b
ca4195e613a44"
VIRUSTOTAL_API_KEY =
"6062658f15f899bc28997f9fdd4e9bb0d12c091b0aea030da57059f73283bb18"

# ARP-сканування
def scan_arp(timeout=3, interface="Ethernet"):
    import psutil
    addrs = psutil.net_if_addrs()
    subnet = "192.168.1.0/24"
    if interface in addrs:
        for snic in addrs[interface]:
            if snic.family.name == 'AF_INET':
                ip = snic.address
                netmask = snic.netmask
                if ip and netmask:
                    import ipaddress
                    net   =   ipaddress.IPv4Network(f"{ip}/{netmask}",
strict=False)
                    subnet = str(net)
                    break
    target_ip = subnet
    print(f"Сканується підмережа: {target_ip}")
    arp = ARP(pdst=target_ip)
    ether = Ether(dst="ff:ff:ff:ff:ff:ff")
    packet = ether / arp
    print("Отримання ARP-таблиці...")
    result    =    srp(packet,    timeout=timeout,    iface=interface,
verbose=0)[0]
    clients = []
    for sent, received in result:
        clients.append({'ip': received.psrc, 'mac': received.hwsrc})
    print(f"Знайдено пристроїв: {len(clients)}")
    return clients
```

```
# Виявлення ARP-спуфінгу
def detect_spoofing(clients):
    mac_table = defaultdict(list)
    for client in clients:
        mac_table[client['mac']].append(client['ip'])
    suspicious = {mac: ips for mac, ips in mac_table.items() if
len(ips) > 1}
    return suspicious

# Сканування з використанням Nmap
def scan_with_nmap(target_subnet="192.168.1.0/24"):
    print(f"Сканування мережі {target_subnet} за допомогою Nmap...")
    scanner = nmap.PortScanner()
    try:
        scanner.scan(hosts=target_subnet, arguments='-sS -T4')
    except Exception as e:
        return {"error": str(e)}

    hosts = []
    for host in scanner.all_hosts():
        host_data = {
            "ip": host,
            "status": scanner[host].state(),
            "open_ports": list(scanner[host]['tcp'].keys()) if 'tcp'
in scanner[host] else []
        }
        hosts.append(host_data)
    return {"scanned_hosts": hosts}

# Аналіз процесів та портів
def scan_process_ports():
    result = []
    for conn in psutil.net_connections(kind='inet'):
        try:
            proc = psutil.Process(conn.pid)
            entry = {
                'pid': conn.pid,
                'name': proc.name(),
                'user': proc.username(),
                'laddr':   f"{conn.laddr.ip}:{conn.laddr.port}"   if
conn.laddr else "",
                'raddr':   f"{conn.raddr.ip}:{conn.raddr.port}"   if
conn.raddr else "",
                'status': conn.status
            }
            result.append(entry)
        except (psutil.AccessDenied, psutil.NoSuchProcess):
            continue
    return result

# Побудова графіка по відкритих портах
```

```
def plot_nmap_ports(nmap_data):
    port_counts = defaultdict(int)
    for host in nmap_data.get("scanned_hosts", []):
        for port in host.get("open_ports", []):
            port_counts[port] += 1

    if not port_counts:
        return None

    ports = list(port_counts.keys())
    counts = list(port_counts.values())

    plt.figure(figsize=(10, 6))
    plt.bar(ports, counts, color="orange")
    plt.title("Найчастіше відкриті порти (Nmap)")
    plt.xlabel("Порт")
    plt.ylabel("Кількість хостів")
    plt.tight_layout()

    image_path = "nmap_ports.png"
    plt.savefig(image_path)
    plt.close()
    return image_path

# Надсилання повідомлення в Telegram
def send_telegram_alert(message):
    url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
    payload = {"chat_id": CHAT_ID, "text": message}
    try:
        response = requests.post(url, data=payload)
        if response.status_code == 200:
            print("Повідомлення відправлено в Telegram.")
        else:
            print("Помилка надсилання:", response.text)
    except Exception as e:
        print("Не вдалося підключитися до Telegram:", e)

# Надсилання зображення в Telegram
def send_telegram_image(image_path):
    url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendPhoto"
    with open(image_path, "rb") as image:
        files = {"photo": image}
        data = {"chat_id": CHAT_ID}
        try:
            response = requests.post(url, files=files, data=data)
            if response.status_code == 200:
                print("Графік відправлено в Telegram.")
            else:
                print("Помилка надсилання графіка:", response.text)
        except Exception as e:
            print("Не вдалося підключитися до Telegram (зображення):",
e)
```

```
# Побудова графіка трафіку
def plot_traffic_graph(ip_counter):
    top_ips = ip_counter.most_common(10)
    ips = [ip for ip, _ in top_ips]
    counts = [count for _, count in top_ips]

    plt.figure(figsize=(10, 6))
    plt.bar(ips, counts, color="skyblue")
    plt.xticks(rotation=45, ha="right")
    plt.title("Топ 10 IP-джерел трафіку")
    plt.xlabel("IP-адреса")
    plt.ylabel("Кількість пакетів")
    plt.tight_layout()

    image_path = "top_traffic.png"
    plt.savefig(image_path)
    plt.close()
    return image_path

# Константи
PACKET_COUNT = 1000
SUSPICIOUS_PACKET_THRESHOLD = 20
LOG_FILE = "network_security_report.json"
SUSPICIOUS_PORTS = {4444, 5555, 6666, 1337, 31337}

# Отримання інформації про IP
def get_ip_info(ip):
    try:
        dns_name = socket.gethostbyaddr(ip)[0]
    except Exception:
        dns_name = "N/A"
    try:
        response = requests.get(f"https://ipinfo.io/{ip}/json")
        data = response.json()
        country = data.get("country", "N/A")
        org = data.get("org", "N/A")
    except Exception:
        country = "N/A"
        org = "N/A"

    abuse_score = "N/A"
    try:
        abuse_headers = {
            "Key": ABUSEIPDB_API_KEY,
            "Accept": "application/json"
        }
        abuse_params = {
            "ipAddress": ip,
            "maxAgeInDays": 30
        }
```

```
        abuse_response                                                =
requests.get("https://api.abuseipdb.com/api/v2/check",
headers=abuse_headers, params=abuse_params)
        if abuse_response.status_code == 200:
            abuse_data = abuse_response.json()
            abuse_score = abuse_data['data']['abuseConfidenceScore']
    except Exception:
        pass

    return {"ip": ip, "dns": dns_name, "country": country, "org": org,
"abuse": abuse_score}

# Вибір інтерфейсу
def choose_interface():
    interfaces = get_working_ifaces()
    print("Доступні інтерфейси:")
    for idx, iface in enumerate(interfaces):
        print(f"[{idx}] {iface.name}")
    try:
        selection = int(input("Введіть номер інтерфейсу: "))
        return interfaces[selection].name
    except (IndexError, ValueError):
        print("Невірний вибір. Обрано перший доступний інтерфейс.")
        return interfaces[0].name if interfaces else None

# Аналіз трафіку
def analyze_traffic(packet_count=PACKET_COUNT, iface=None):
    print(f"Захоплення  {packet_count}  IP-пакетів  на  інтерфейсі
{iface}...")
    packets  =  sniff(count=packet_count,  iface=iface,  filter="ip",
timeout=10)
    ip_counter = Counter()
    for pkt in packets:
        if IP in pkt:
            src_ip = pkt[IP].src
            ip_counter[src_ip] += 1

    suspicious_ips = {ip: count for ip, count in ip_counter.items()
if count >= SUSPICIOUS_PACKET_THRESHOLD}
    ip_details = {ip: get_ip_info(ip) for ip in suspicious_ips}

    return {
        "total_packets": len(packets),
        "top_sources": ip_counter.most_common(10),
        "suspicious_ips": suspicious_ips,
        "ip_info": ip_details
    }

# Генерація звіту
def  save_report(traffic_data,  nmap_data,  arp_clients,  spoofed,
proc_data, filename="security_report.txt"):
    now = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
    with open(filename, "w", encoding="utf-8") as f:
        f.write(f"Звіт безпеки - {now}\n")
        f.write("=" * 60 + "\n\n")

        f.write("ARP-таблиця:\n")
        for client in arp_clients:
            f.write(f"IP: {client['ip']}\tMAC: {client['mac']}\n")

        f.write("\nARP-спуфінг:\n")
        if spoofed:
            for mac, ips in spoofed.items():
                f.write(f"MAC {mac} використовується кількома IP: {',
'.join(ips)}\n")
        else:
            f.write("Підозрілих MAC-адрес не виявлено\n")

        f.write("\nАктивні мережеві процеси:\n")
        for proc in proc_data:
            f.write(f"PID:   {proc['pid']}   |   {proc['name']}   |
Користувач: {proc['user']} | {proc['laddr']} -> {proc['raddr']} |
{proc['status']}\n")

        f.write("\nІнформація про підозрілі IP:\n")
        for ip, info in traffic_data.get("ip_info", {}).items():
            f.write(f"IP:  {ip}  |  DNS:  {info['dns']}  |  Країна:
{info['country']}  |  Організація:  {info['org']}  |  Abuse  Score:
{info['abuse']}\n")

# Запуск основної програми
if __name__ == "__main__":
    iface = choose_interface()
    if not iface:
        print("Не знайдено активних мережевих інтерфейсів.")
        exit(1)

    traffic_data = analyze_traffic(iface=iface)
    arp_clients = scan_arp(interface=iface)
    spoofed = detect_spoofing(arp_clients)
    proc_data = scan_process_ports()
    nmap_data = scan_with_nmap()
    img_nmap = plot_nmap_ports(nmap_data)
    if img_nmap:
        send_telegram_image(img_nmap)

    report_file = "security_report.txt"
    save_report(traffic_data,   nmap_data,   arp_clients,   spoofed,
proc_data, filename=report_file)

    if traffic_data["suspicious_ips"]:
        alert_msg = "Виявлено підозрілу активність у мережі.\n"
        alert_msg      +=      f"Загальна      кількість      пакетів:
{traffic_data['total_packets']}\n"
```

```
        alert_msg += f"Найактивніші джерела трафіку:\n"
        for ip, count in traffic_data['top_sources'][:5]:
            alert_msg += f"- {ip}: {count} пакетів\n"
        alert_msg += "\nДеталі підозрілих IP-адрес:\n"
        for ip, count in traffic_data["suspicious_ips"].items():
            info = traffic_data["ip_info"].get(ip, {})
            alert_msg += (
                f"IP: {ip} - {count} пакетів\n"
                f"DNS:       {info.get('dns')}       |       Країна:
{info.get('country')} | Організація: {info.get('org')}\n"
                f"Abuse:     {info.get('abuse')}     |     VirusTotal:
{info.get('vt', 'N/A')}\n\n"
            )
        send_telegram_alert(alert_msg)

        # Побудова графіка та надсилання в Telegram
        img_path                                                    =
plot_traffic_graph(Counter(dict(traffic_data['top_sources'])))
        send_telegram_image(img_path)

    summary = (
        "Перевірка завершена.\n"
        f"Захоплено пакетів: {traffic_data['total_packets']}\n"
        f"Підозрілих                                          IP-адрес:
{len(traffic_data['suspicious_ips'])}\n"
        f"Кількість               відсканованих              хостів:
{len(nmap_data.get('scanned_hosts', []))}\n"
    )
    send_telegram_alert(summary)
```