МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА УПРАВЛІННЯ КІБЕРБЕЗПЕКОЮ ТА ЗАХИСТОМ ІНФОРМАЦІЇ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: “МЕТОДИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ЖУРНАЛІВ ПОДІЙ ТА ДОКАЗОВОЇ БАЗИ В СИСТЕМАХ КІБЕРБЕЗПЕКИ ОРГАНІЗАЦІЇ”

на здобуття освітнього ступеня бакалавра
зі спеціальності 125 Кібербезпека
освітньої програми Управління інформаційною та кібернетичною безпекою

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олексій ЮРЧИШИН
(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. УБД-42

Олексій ЮРЧИШИН
Ім’я, ПРІЗВИЩЕ

Керівник: Юрій ЯКИМЕНКО
Ім’я, ПРІЗВИЩЕ

Рецензент: Сергій ГАХОВ
Ім’я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут кібербезпеки та захисту інформації

Кафедра управління кібербезпекою та захистом інформації

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека

Освітня програма Управління інформаційною та кібернетичною безпекою

ЗАТВЕРДЖУЮ
Завідувач кафедри УКБЗІ
_____ Світлана ЛЕГОМІНОВА
“____”_____2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Юрчишин Олексій Михайлович
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи “Методи забезпечення цілісності журналів подій та доказової бази в системах кібербезпеки організації”,
Керівник кваліфікаційної роботи Юрій ЯКИМЕНКО, доцент каф., к. в. н., доцент
(ІПРІЗВИЩЕ, Ім’я,, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20“ лютого 2026 р. №51.

2. Строк подання кваліфікаційної роботи “12” травня 2026р.

3. Вихідні дані до кваліфікаційної роботи: *системи кібербезпеки організації, журнали подій (логи), міжнародні стандарти у сфері інформаційної безпеки, науково-технічна література з питань забезпечення цілісності даних та формування доказової бази*.

4. Перелік питань, які мають бути розроблені:
4.1. Дослідити проблеми забезпечення цілісності журналів подій у системах кібербезпеки організацій.
4.2. Проаналізувати методи та засоби контролю цілісності логів і формування достовірної доказової бази відповідно до міжнародних стандартів інформаційної безпеки.
4.3. Розробити та реалізувати систему захисту журналів подій.

5. Перелік ілюстративного матеріалу: *презентація PowerPoint*

6. Дата видачі завдання “05“ березня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Визначення об’єкта, предмета, мети та завдань дослідження у сфері забезпечення цілісності журналів подій	18.03.2026	
2.	Збір та аналіз наукової літератури і нормативних джерел щодо захисту логів та доказової інформації	29.03.2026	
3.	Дослідження міжнародних стандартів і вимог до забезпечення цілісності журналів подій та ведення аудиту	08.04.2026	
4.	Аналіз існуючих підходів і політик забезпечення цілісності логів у системах кібербезпеки організацій	22.04.2026	
5.	Оцінка відповідності механізмів захисту журналів подій міжнародним стандартам та виявлення вразливостей	08.05.2026	
6.	Розробка та реалізація системи захисту журналів подій.	20.05.2026	
7.	Оформлення кваліфікаційної роботи	22.05.2026	
8.	Підготовка та оформлення презентаційних матеріалів	03.06.2026	
9.	Отримання рецензії на роботу	08.06.2026	
10.	Захист кваліфікаційної роботи в ЕК	__.06.2026	

Здобувач вищої освіти _____ Олексій ЮРЧИШИН
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Юрій ЯКИМЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра

Направляється здобувач Юрчишин О.М. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 125 Кібербезпека
(код, найменування спеціальності)
освітньої програми Управління інформаційною та кібернетичною безпекою
(назва)
на тему: "Методи забезпечення цілісності журналів подій та доказової бази в системах кібербезпеки організації"

Кваліфікаційна робота і рецензія додаються.

Директор ННІКБЗІ _____ Євгенія ІВАНЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач ЮРЧИШИН Олексій у кваліфікаційній роботі вивчив особливості журналювання подій та збереження даних як доказової бази в системах кібербезпеки, дослідив загрози порушення цілісності логів, проаналізував ефективності використання ефективність криптографічних методів і механізмів аудиту для їхнього захисту, розробив рекомендації та технічні рішення темою дослідження.

ЮРЧИШИН Олексій показав розуміння проблеми дослідження та бачення основних теоретичних і практичних напрямів її вирішення, довів володіння методами наукового дослідження, проявив себе як організований, відповідальний виконавець. Результати дослідження апробовані на конференції.

Все це дозволяє оцінити кваліфікаційну роботу здобувача ЮРЧИШИНА Олексія на оцінку "добре" та присвоїти йому кваліфікацію бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Керівник кваліфікаційної роботи _____ Юрій ЯКИМЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

"____"_____2026 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Юрчишин О.М. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри
управління кібербезпекою та
захистом інформації _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну бакалаврську роботу

здобувача вищої освіти ЮРЧИШИНА Олексія
на тему “Методи забезпечення цілісності журналів подій та доказової бази в системах кібербезпеки організації”

Актуальність. У сучасному цифровому світі, де кіберзагрози стають дедалі витонченішими, надійні журнали подій залишаються основним інструментом виявлення кібератак. Державні установи та корпорації дедалі частіше стикаються з атаками, де зловмисники намагаються приховати свої сліди шляхом модифікації, підробки або видалення системних логів. З огляду на ризики втрати критично важливих даних, надзвичайно важливо розробляти методи криптографічного контролю та впроваджувати надійні механізми перевірки цілісності.

Дослідження методів захисту журналів подій та створення практичних рекомендацій щодо формування доказової бази є актуальним науково-прикладним завданням у сфері інформаційної безпеки.

Позитивні сторони.

1. У роботі проведено глибокий аналіз проблем забезпечення цілісності журналів подій та особливостей формування достовірної доказової бази.
2. Кваліфікаційна робота оформлена відповідно до вимог. Виклад матеріалу здійснено відповідно до плану, зроблено логічні висновки. Ключові положення роботи представлено у вигляді рисунків і таблиць.
3. Автор опрацював значну джерельну базу: понад 30 публікацій, в тому числі міжнародні стандарти та англомовні матеріали.
4. За результатами дослідження запропоновано практичні рекомендації та алгоритми та методики для захисту логів від несанкціонованої модифікації.

Недоліки.

Доцільно було б приділити більше уваги особливостям забезпечення цілісності журналів подій у хмарних та гібридних інфраструктурах, оскільки процеси збору, передачі та зберігання логів у таких середовищах мають свою специфіку порівняно з локальними мережами.

Однак, вищезгадані зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Кваліфікаційна робота виконана на належному науково-методичному рівні і заслуговує оцінки “добре”, а здобувач ЮРЧИШИН Олексій заслуговує присвоєння кваліфікації бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Рецензент: _____ Сергій ГАХОВ
к.в.н., доцент *підпис* Ім’я, ПРІЗВИЩЕ

РЕФЕРАТ

Кваліфікаційна робота присвячена дослідженню методів забезпечення цілісності журналів подій у системах кібербезпеки організації, аналізу загроз порушення достовірності логів та формування на їх основі доказової бази, а також розробці рекомендацій щодо вдосконалення механізмів захисту, зберігання та перевірки журналів подій із використанням сучасних криптографічних і аудиторських підходів. Робота складається зі вступу, трьох розділів, що містять 20 рисунків, висновківі списку використаних джерел із 32 найменування. Загальний обсяг роботи становить 65 аркушів, з яких 4 аркуші займають перелік умовних скорочень та список використаних джерел.

Метою роботи є розробка методів та програмних засобів забезпечення цілісності журналів подій, що дозволяють запобігти їх несанкціонованій зміні, забезпечити достовірність даних та використовувати їх як доказову базу при розслідуванні інцидентів.

Об'єктом дослідження є процеси формування, збереження та аналізу журналів подій у системах кібербезпеки організації.

Предмет дослідження – методи та засоби забезпечення цілісності журналів подій і формування доказової бази для виявлення інцидентів інформаційної безпеки.

Методи дослідження.

1. Аналіз сучасних підходів до ведення журналів подій у системах кібербезпеки.
2. Дослідження загроз та вразливостей, пов'язаних із порушенням цілісності логів.
3. Проєктування архітектури системи захисту журналів подій.
4. Використання криптографічних методів (хешування, цифрові підписи).
5. Реалізація механізмів контролю змін та аудиту подій.

Галузь застосування. Розроблені підходи можуть бути використані для забезпечення цілісності журналів подій у системах кібербезпеки організацій, підвищення достовірності логів як доказової бази при розслідуванні інцидентів, а також для вдосконалення процесів аудиту, моніторингу та управління інформаційною безпекою із застосуванням сучасних методів контролю та криптографічного захисту.

Ключові слова: ЖУРНАЛИ ПОДІЙ, КІБЕРБЕЗПЕКА, ЦІЛІСНІСТЬ ДАНИХ, ХЕШУВАННЯ, ЦИФРОВИЙ ПІДПИС, ЛОГУВАННЯ, АУДИТ, ДОКАЗОВА БАЗА, ІНЦИДЕНТИ БЕЗПЕКИ

ABSTRACT

The qualification work is dedicated to the research of methods for ensuring the integrity of event logs in the cybersecurity systems of an organization, the analysis of threats of violation of the authenticity of logs and the formation of an evidence base on their basis, as well as the development of recommendations regarding the improvement of mechanisms for the protection, storage, and verification of event logs using modern cryptographic and auditing approaches. The work consists of an introduction, three chapters that contain 20 figures, conclusions, and a list of used sources of 32 items. The total volume of the work constitutes 65 sheets, of which 4 sheets are occupied by the list of conventional abbreviations and the list of used sources.

The purpose of the study is the development of methods and software tools for ensuring the integrity of event logs, which allow to prevent their unauthorized modification, to ensure the authenticity of data, and to use them as an evidence base during the investigation of incidents.

The object of the study is the processes of formation, storage, and analysis of event logs in the cybersecurity systems of an organization.

The subject study is the methods and tools for ensuring the integrity of event logs and the formation of an evidence base for the detection of information security incidents.

Research methods.

1. Analysis of modern approaches to maintaining event logs in cybersecurity systems.
2. Research of threats and vulnerabilities related to the violation of the integrity of logs.
3. Design of the architecture of the event log protection system.
4. Use of cryptographic methods (hashing, digital signatures).
5. Implementation of change control and event audit mechanisms.

***Field of application*.** The developed approaches can be used for ensuring the integrity of event logs in the cybersecurity systems of organizations, increasing the authenticity of logs as an evidence base during the investigation of incidents, as well as for the improvement of processes of audit, monitoring, and management of information security with the application of modern methods of control and cryptographic protection.

Keywords: EVENT LOGS, CYBERSECURITY, DATA INTEGRITY, HASHING, DIGITAL SIGNATURE, LOGGING, AUDIT, EVIDENCE BASE, SECURITY INCIDENTS.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасних умовах стрімкого розвитку інформаційних технологій та зростання кількості кіберзагроз особливої актуальності набуває забезпечення надійного функціонування систем кібербезпеки організацій. Одним із ключових елементів таких систем є журнали подій (логи), які використовуються для моніторингу діяльності користувачів, виявлення інцидентів безпеки та проведення їх подальшого розслідування. Саме на основі логів формується доказова база, що дозволяє відновити послідовність подій та встановити факт порушення безпеки.

Разом із тим, журнали подій часто стають об’єктом атак з боку зловмисників, які намагаються змінити, підробити або видалити записи з метою приховування своїх дій. Це призводить до втрати достовірності інформації, ускладнює процес аналізу інцидентів і знижує ефективність систем захисту. Недостатній рівень забезпечення цілісності логів може зробити їх непридатними як доказову базу, зокрема у випадках внутрішніх розслідувань або судових процесів.

У зв’язку з цим особливої важливості набуває розробка та впровадження ефективних методів забезпечення цілісності журналів подій, що включають використання криптографічних механізмів, систем аудиту та контролю змін. Важливим є також врахування міжнародних стандартів у сфері інформаційної безпеки, які регламентують вимоги до ведення та захисту логів.

Таким чином, дослідження методів забезпечення цілісності журналів подій та формування на їх основі достовірної доказової бази є актуальним науково-практичним завданням, спрямованим на підвищення рівня кібербезпеки організацій та ефективності реагування на інциденти інформаційної безпеки.

Метою роботи є розробка методів та програмних засобів забезпечення цілісності журналів подій, що дозволяють запобігти їх несанкціонованій зміні, забезпечити достовірність даних та використовувати їх як доказову базу при розслідуванні інцидентів.

Об'єктом дослідження є процеси формування, збереження та аналізу журналів подій у системах кібербезпеки організації.

Предмет дослідження – методи та засоби забезпечення цілісності журналів подій і формування доказової бази для виявлення інцидентів інформаційної безпеки.

Для досягнення цієї мети в роботі необхідно виконати наступні завдання:

1. Проаналізувати сучасні підходи до ведення журналів подій у системах кібербезпеки організацій.
2. Дослідити роль журналів подій у виявленні інцидентів інформаційної безпеки та формуванні доказової бази.
3. Визначити основні загрози та вразливості, пов'язані з порушенням цілісності логів.
4. Дослідити існуючі методи забезпечення цілісності логів, зокрема криптографічні підходи (хешування, цифровий підпис).
5. Розробити архітектуру системи забезпечення цілісності журналів подій та їх захищеного зберігання.
6. Запропонувати механізми контролю змін і перевірки достовірності логів.
7. Реалізувати підсистему збору, зберігання та захисту журналів подій.
8. Провести аналіз ефективності запропонованих рішень у контексті формування доказової бази.
9. Сформувати практичні рекомендації щодо вдосконалення процесів забезпечення цілісності логів у системах кібербезпеки організацій.

Методи дослідження:

1. Аналіз сучасних підходів до ведення журналів подій у системах кібербезпеки.
2. Дослідження загроз та вразливостей, пов'язаних із порушенням цілісності логів.
3. Проєктування архітектури системи захисту журналів подій.
4. Використання криптографічних методів (хешування, цифрові

підписи).

5. Реалізація механізмів контролю змін та аудиту подій.

Практичне значення одержаних результатів. Застосування розроблених підходів дозволить організаціям забезпечити надійний контроль цілісності журналів подій, підвищити достовірність логів як джерела доказової бази при розслідуванні інцидентів інформаційної безпеки, а також удосконалити процеси моніторингу, аудиту та управління кібербезпекою. Це сприятиме своєчасному виявленню порушень, запобіганню несанкціонованим змінам даних та загальному підвищенню рівня захисту інформаційних ресурсів організації.

Апробація результатів кваліфікаційної роботи відбулася на Всеукраїнській науково-практичній конференції “Стратегії кіберстійкості управління ризикаи та безперервінсть бізнесу” 25 лютого 2026 року.

РОЗДІЛ 1 АНАЛІЗ ВПЛИВУ ЖУРНАЛЮВАННЯ І ЗБЕРЕЖЕННЯ ДАНИХ, ЯК ДОКАЗОВОЇ БАЗИ, В СИСТЕМАХ КІБЕРБЕЗПЕКИ

1.1 Роль журналів подій інформаційної безпеки у кібербезпеці

У інформаційних системах журналювання подій є невід'ємною складовою забезпечення кібербезпеки, оскільки саме за допомогою цього механізму здійснюється фіксація всіх значущих дій, що відбуваються в системі. Логування виступає базовим інструментом контролю, який дозволяє відстежувати поведінку користувачів, стан програмних компонентів та реакцію системи на зовнішні й внутрішні впливи.

Журналюванням подій – це процес автоматичного запису інформації про події, що відбуваються в інформаційній системі, у спеціалізовані журнали (логи). До таких подій можуть належати спроби автентифікації, доступ до ресурсів, виконання операцій, зміни конфігурації, системні помилки, а також дії, що можуть свідчити про порушення безпеки [1]. Важливо, що логування не обмежується лише фіксацією фактів, а формує структурований набір даних, який може бути використаний для подальшого аналізу та прийняття управлінських рішень у сфері інформаційної безпеки.

Процес журналювання передбачає кілька взаємопов'язаних етапів. На першому етапі відбувається генерація події, яка формується внаслідок дії користувача або роботи системного компонента. Далі здійснюється її обробка та нормалізація, що включає визначення типу події, її класифікацію та доповнення метаданими, такими як час, ідентифікатор користувача, джерело події тощо. Після цього інформація записується у відповідний журнал, який може зберігатися локально або передаватися до централізованої системи збору логів. Завершальним етапом є використання накопичених даних для моніторингу, аналізу та виявлення інцидентів безпеки.

Особливого значення журналювання набуває у контексті забезпечення кібербезпеки організацій, оскільки воно дозволяє реалізувати принципи

підзвітності та прозорості дій у системі. Завдяки логам стає можливим відновлення послідовності подій, що передували інциденту, визначення джерела загрози та оцінка масштабу порушення. Крім того, журнали подій використовуються для виявлення аномальної активності, що може свідчити про спроби несанкціонованого доступу або проведення атак.

Журнали подій активно використовуються в системах класу SIEM (Security Information and Event Management), які забезпечують збір, кореляцію та аналіз великого обсягу даних з різних джерел у режимі реального часу. Завдяки цьому стає можливим виявлення складних атак, які неможливо ідентифікувати на основі окремих подій, оскільки SIEM-системи дозволяють встановлювати взаємозв’язки між різними записами логів та формувати цілісну картину інциденту [2].

Окрім цього, журнали подій є основним джерелом інформації для функціонування центрів операційної безпеки (SOC — Security Operations Center), діяльність яких спрямована на безперервний моніторинг стану інформаційної інфраструктури організації. Аналітики SOC використовують лог-файли для виявлення підозрілої активності, аналізу інцидентів та прийняття рішень щодо реагування. Саме завдяки журналам подій забезпечується можливість оперативного реагування на кіберзагрози та мінімізації їх наслідків.

Крім того, логування широко застосовується в системах моніторингу, які дозволяють відстежувати стан програмного та апаратного забезпечення, виявляти збої, перевантаження або аномалії в роботі системи. У цьому контексті журнали подій виконують роль інформаційної основи для оцінки працездатності інфраструктури та забезпечення її стабільного функціонування. Вони дають змогу не лише реагувати на вже наявні проблеми, а й прогнозувати потенційні загрози на основі аналізу тенденцій і поведінкових патернів (рис. 1.1).

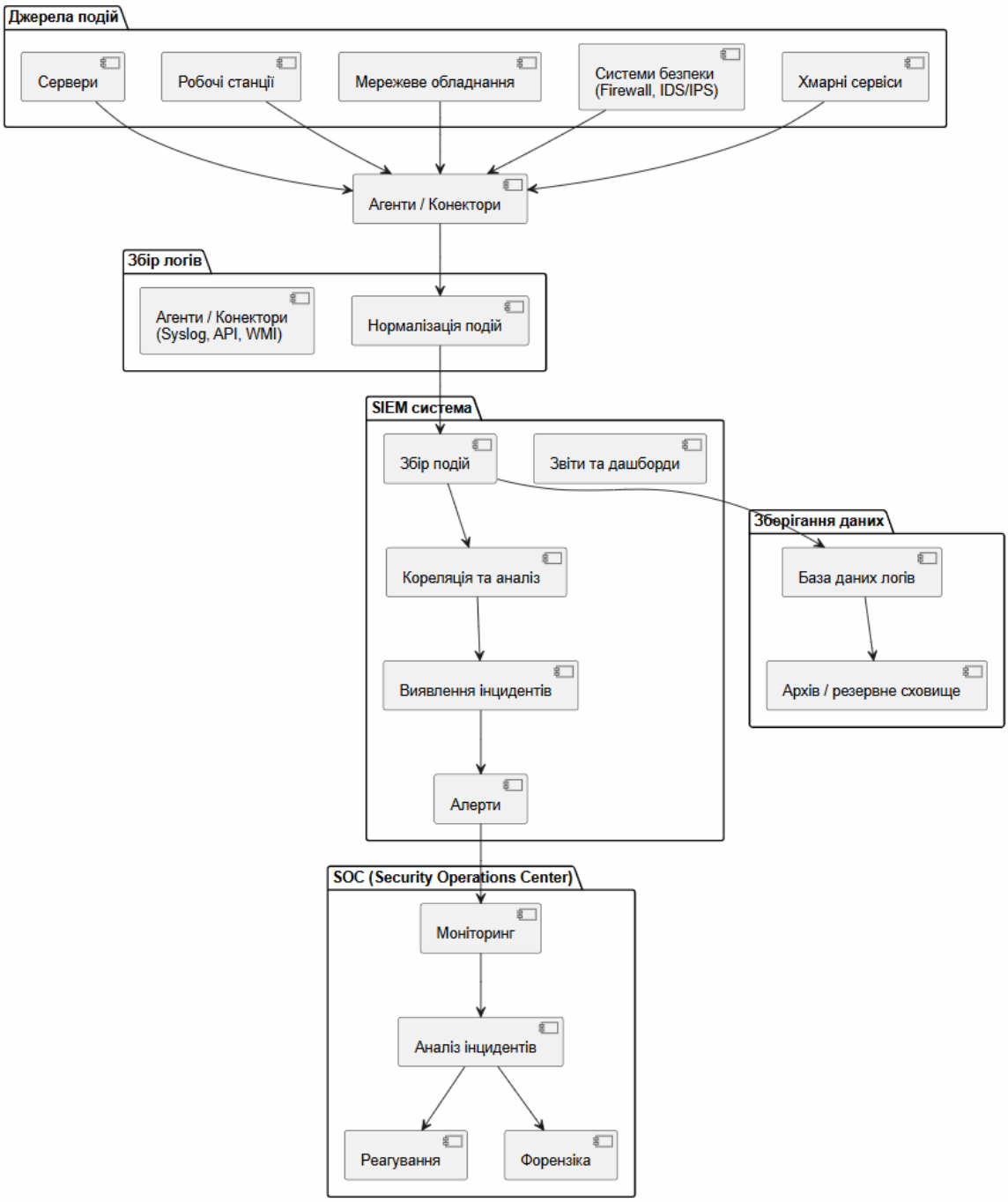

Рис. 1.1 Архітектура SIEM/SOC та процес журналювання подій

Використання журналів подій у системах SIEM, центрах SOC та засобах моніторингу підтверджує їх ключову роль у забезпеченні комплексного підходу до кібербезпеки, що включає виявлення, аналіз і реагування на інциденти в інформаційних системах організації.

У процесі журналювання подій фіксується широкий спектр інформації, що відображає як звичайне функціонування системи, так і потенційно небезпечну

активність. Насамперед у журналах подій реєструються дані про входи до системи, зокрема спроби автентифікації користувачів, причому як успішні, так і невдалі. Така інформація дає змогу відстежувати факти доступу до інформаційних ресурсів, виявляти несанкціоновані спроби входу та формувати уявлення про характер використання системи.

Водночас елементом журналювання є фіксація дій користувачів після входу до системи. До таких дій належать операції з файлами, зміна конфігураційних параметрів, виконання програм, доступ до баз даних та інші активності, що можуть впливати на стан інформаційної інфраструктури. Саме ці записи дозволяють відновити послідовність подій у разі виникнення інциденту, забезпечуючи можливість детального аналізу та подальшого прийняття обґрунтованих рішень у сфері кібербезпеки [4].

Паралельно з цим у журналах подій реєструються системні помилки та збої, які можуть бути як наслідком технічних несправностей, так і ознакою спроб несанкціонованого втручання. Їх своєчасне виявлення дозволяє не лише підвищити надійність роботи системи, але й запобігти розвитку потенційних загроз. У сукупності з цим особливе значення має фіксація подій, що свідчать про можливі атаки або аномальну поведінку, зокрема спроби підбору паролів, перевищення прав доступу, підозрілі запити або інші відхилення від нормального режиму функціонування.

Таким чином, журналювання подій забезпечує комплексне відображення діяльності інформаційної системи, поєднуючи дані про доступ, дії користувачів, технічні стани та безпекові інциденти. Саме така повнота та взаємозв’язок інформації створюють основу для ефективного моніторингу, аналізу та формування достовірної доказової бази, що є критично важливим для забезпечення кібербезпеки організації.

Отже, журнали подій виступають ключовим джерелом інформації про функціонування інформаційної системи та дії її користувачів, забезпечуючи можливість відстеження, аналізу та виявлення інцидентів кібербезпеки. Разом із тим їх значення не обмежується лише технічним моніторингом, оскільки

накопичені дані можуть використовуватись як основа для формування доказової бази під час проведення аудиту, розслідування інцидентів та прийняття управлінських рішень.

У зв’язку з цим виникає необхідність більш детального розгляду поняття доказової бази в системах кібербезпеки, її ролі в управлінських процесах та вимог до забезпечення достовірності й цілісності інформації, що в ній використовується.

1.2 Призначення доказової бази в управлінських процесах забезпечення інформаційної безпеки

У сучасних умовах функціонування інформаційних систем важливого значення набуває формування та використання достовірної доказової бази, яка забезпечує обґрунтованість прийняття рішень у сфері інформаційної безпеки. Під доказовою базою у контексті кібербезпеки розуміють сукупність зафіксованих, перевірених та належним чином збережених даних, що відображають факти подій, дій користувачів або станів системи і можуть бути використані для підтвердження або спростування певних тверджень щодо інцидентів безпеки.

Формування доказової бази тісно пов’язане з процесом журналювання подій, оскільки саме журнали виступають основним джерелом первинної інформації. Проте не кожен запис у журналі автоматично може вважатися доказом [5]. Для цього інформація повинна відповідати ряду вимог, зокрема бути достовірною, повною, актуальною та захищеною від несанкціонованих змін. Лише за таких умов вона може використовуватися як підстава для прийняття управлінських рішень, проведення аудиту або розслідування інцидентів.

Процес формування доказової бази в системах кібербезпеки є поетапним і передбачає не лише збір інформації, а й її подальшу обробку, перевірку та збереження. На початковому етапі здійснюється збір даних із різних джерел, серед яких основне місце займають журнали подій, а також мережеві журнали,

системи контролю доступу, засоби моніторингу та інші компоненти інформаційної інфраструктури. Важливо, що на цьому етапі забезпечується повнота та безперервність фіксації подій, оскільки навіть незначні пропуски можуть вплинути на цілісність подальшого аналізу.

Наступним етапом є обробка та структурування отриманої інформації, що включає її нормалізацію, класифікацію та виділення релевантних подій. У цей момент відбувається відбір даних, які можуть мати доказове значення, а також їх упорядкування у вигляді, придатному для подальшого використання. Особлива увага приділяється встановленню часових зв’язків між подіями, що дозволяє відтворити послідовність дій та сформувати цілісну картину інциденту.

Ключовим етапом формування доказів є забезпечення їх достовірності та цілісності. Для цього застосовуються спеціальні методи, зокрема криптографічне хешування, цифрові підписи, контроль доступу та механізми аудиту, які унеможливлюють несанкціоновану зміну або підробку даних. Завдяки цьому кожен запис, що входить до доказової бази, може бути перевірений на предмет незмінності та відповідності початковому стану.

Завершальним етапом є зберігання та використання сформованої доказової бази. Дані повинні зберігатися у захищеному середовищі з дотриманням вимог безпеки, що гарантує їх доступність для подальшого аналізу, аудиту або розслідування інцидентів [6]. У результаті формується структурована та перевірена інформація, яка може бути використана як на технічному рівні, так і в управлінських або юридичних процесах.

Важливість доказової бази в системах кібербезпеки значною мірою визначається роллю журналів подій у різних управлінських та прикладних процесах. Зокрема, у контексті управління інцидентами інформаційної безпеки журнали подій виступають основним джерелом інформації для виявлення, аналізу та реагування на загрози. Саме на основі логів здійснюється ідентифікація інциденту, визначення його джерела, часу виникнення та послідовності дій, що передували порушенню безпеки. Це дозволяє оперативно приймати рішення щодо локалізації інциденту, мінімізації його наслідків та

запобігання повторним атакам.

Логи забезпечують можливість перевірки дотримання політик безпеки, контролю доступу до інформаційних ресурсів та відповідності діяльності користувачів встановленим правилам. Завдяки аналізу журналів подій аудитори можуть виявляти порушення, оцінювати ефективність впроваджених заходів захисту та формувати обґрунтовані висновки щодо стану системи безпеки організації [7]. У цьому контексті журнали подій виступають інструментом підзвітності та прозорості функціонування інформаційної інфраструктури.

У разі виникнення конфліктних ситуацій або правопорушень саме лог-файли можуть бути використані як доказ факту здійснення певних дій або настання події. Однак для цього необхідно, щоб вони відповідали встановленим вимогам до доказової інформації, зокрема щодо цілісності, достовірності та захищеності від змін. Лише за умови належного забезпечення цих характеристик журнали подій можуть бути визнані допустимими доказами в межах юридичних процедур.

Враховуючи значення журналів подій у процесах управління інцидентами, аудиту та юридичного підтвердження фактів порушення безпеки, особливої уваги потребують вимоги до формування доказової бази. Насамперед така інформація повинна відповідати критерію цілісності, що передбачає відсутність несанкціонованих змін у даних з моменту їх створення. Порушення цілісності логів призводить до втрати довіри до інформації та унеможливлює її використання як доказу.

Не менш важливою характеристикою є достовірність, яка визначає відповідність зафіксованих даних реальним подіям, що відбулися в системі. Забезпечення достовірності передбачає використання надійних механізмів збору інформації, синхронізації часу, а також контроль джерел формування подій. Лише за умови достовірності дані можуть бути використані для прийняття управлінських або юридично значущих рішень.

Крім того, ключовою вимогою до доказової інформації є її незмінність, яка гарантує, що після фіксації подій дані не можуть бути змінені або підроблені без

можливості виявлення таких змін. Це досягається шляхом застосування спеціалізованих механізмів захисту, зокрема криптографічних методів, систем аудиту та контролю доступу.

Дотримання вимог цілісності, достовірності та незмінності є необхідною умовою формування якісної доказової бази, яка може ефективно використовуватися в системах кібербезпеки організації. Для наочного відображення ключових характеристик доказової бази доцільно подати їх у вигляді структурної схеми, яка демонструє взаємозв'язок між основними вимогами до доказових даних у системах кібербезпеки (рис. 1.2).

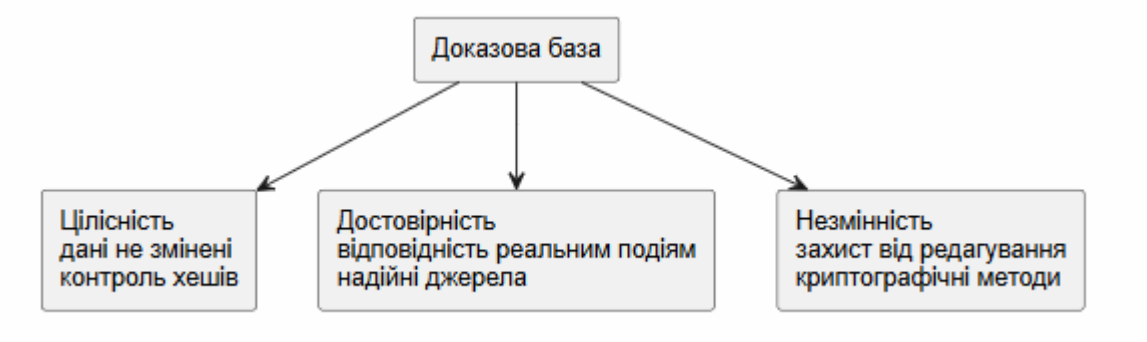

Рис. 1.2 Основні вимоги до доказової бази в системах кібербезпеки

Висновок до розділу 1

Розглянуто теоретичні аспекти використання журналів подій та механізмів збереження даних у системах кібербезпеки. Досліджено роль журналювання як одного з ключових інструментів контролю, моніторингу та фіксації подій інформаційної безпеки. Визначено, що журнали подій забезпечують можливість реєстрації дій користувачів, системних процесів, спроб несанкціонованого доступу та інших інцидентів, що є важливим для своєчасного виявлення кіберзагроз і реагування на них.

Проаналізовано основні типи журналів подій, принципи їх формування та значення у сучасних системах інформаційної безпеки. Встановлено, що ефективне журналювання дозволяє підвищити рівень контролю над

інформаційною інфраструктурою, забезпечити прозорість роботи системи та створити основу для подальшого аналізу інцидентів. Також було визначено, що централізоване зберігання журналів і використання автоматизованих засобів моніторингу сприяють підвищенню ефективності систем кіберзахисту.

Досліджено призначення доказової бази в управлінських процесах забезпечення інформаційної безпеки. Встановлено, що журнали подій та збережені цифрові артефакти можуть використовуватися як доказова база під час розслідування інцидентів, аудиту безпеки та прийняття управлінських рішень. Окрему увагу приділено питанням цілісності, достовірності та захищеності даних, оскільки саме ці характеристики визначають можливість використання інформації як надійного доказу у процесах аналізу кіберінцидентів.

РОЗДІЛ 2 РОЗРОБКА ПІДХОДІВ ДО ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ЖУРНАЛІВ ПОДІЙ ТА ДОКАЗОВОЇ БАЗИ В СИСТЕМАХ КІБЕРБЕЗПЕКИ ОРГАНІЗАЦІЇ

У другому розділі кваліфікаційної роботи розглядаються практичні та організаційно-технічні підходи до забезпечення цілісності журналів подій і доказової бази в системах кібербезпеки організації. Основну увагу приділено питанням побудови механізмів захисту журналів подій від несанкціонованої модифікації, втрати або підробки, а також способам забезпечення достовірності та надійності збережених даних.

У межах розділу буде розкрито принципи організації процесів журналювання, підходи до централізованого зберігання та контролю цілісності даних, а також методи використання криптографічних засобів захисту для забезпечення незмінності журналів подій. Крім того, буде розглянуто архітектуру системи зберігання доказової бази, механізми контролю доступу, резервного копіювання та аудиту подій інформаційної безпеки.

2.1 Аналіз вимог до логування подій інформаційної безпеки

Ефективне забезпечення цілісності журналів подій та формування достовірної доказової бази в системах кібербезпеки організації безпосередньо залежить від правильності організації процесу логування та дотримання встановлених вимог до збору, обробки й зберігання інформації. У сучасних умовах зростання кількості кіберзагроз і ускладнення інформаційних інфраструктур логування перестає бути допоміжним механізмом і перетворюється на один із ключових елементів системи захисту, що забезпечує не лише моніторинг подій, але й можливість їх подальшого аналізу, аудиту та використання як доказової бази [8]. Неналежна організація логування, зокрема відсутність повноти фіксації подій, некоректна синхронізація часу, недостатній рівень захисту або відсутність централізованого зберігання, може призвести до

втрати важливої інформації, спотворення даних або неможливості їх використання у процесах розслідування інцидентів.

З огляду на це особливої актуальності набуває визначення та аналіз вимог до логування подій інформаційної безпеки, які повинні враховувати як технічні аспекти функціонування системи, так і організаційні та нормативні обмеження. Такі вимоги формуються на основі міжнародних стандартів і практик у сфері інформаційної безпеки, зокрема рекомендацій щодо ведення журналів подій, забезпечення їх цілісності, захищеності та доступності для аналізу. Вони охоплюють питання повноти збору інформації, точності фіксації подій, захисту даних від несанкціонованого доступу та змін, а також можливості їх використання для виявлення інцидентів, проведення аудиту та формування доказової бази.

Після визначення загальних вимог до логування подій інформаційної безпеки доцільно детальніше розглянути функціональні вимоги, які безпосередньо визначають можливості системи щодо збору, зберігання та використання журналів подій. Саме функціональні вимоги формують основу для проєктування системи логування, оскільки вони визначають, які саме процеси повинні бути реалізовані для забезпечення ефективного контролю, аналізу та подальшого використання даних у системах кібербезпеки організації.

Однією з ключових функціональних вимог є забезпечення повноцінного збору логів з різних джерел інформаційної інфраструктури. Система повинна підтримувати отримання даних із серверів, робочих станцій, мережевого обладнання, систем безпеки, а також хмарних сервісів [9]. При цьому важливо забезпечити безперервність збору інформації, її уніфікацію та можливість обробки подій різних форматів. Ефективний збір логів дозволяє сформувати цілісну картину функціонування системи та забезпечує основу для подальшого аналізу подій.

Не менш важливою є функція збереження журналів подій, яка передбачає надійне та захищене зберігання даних протягом визначеного періоду часу. Система повинна забезпечувати централізоване зберігання логів, їх

резервування, а також захист від втрати або пошкодження інформації. Особливу увагу необхідно приділяти питанням цілісності та незмінності даних, що досягається шляхом застосування механізмів контролю змін, розмежування доступу та використання криптографічних методів.

Крім того, важливою функціональною вимогою є забезпечення доступу до журналів подій, який має бути організований відповідно до принципів безпеки та контролю прав користувачів. Система повинна надавати можливість швидкого пошуку, фільтрації та аналізу логів, а також формування звітів для різних категорій користувачів. Водночас доступ до даних має бути обмежений відповідно до ролей і повноважень, що дозволяє запобігти несанкціонованому використанню або зміні інформації.

Для систематизації функціональних вимог до логування подій доцільно подати їх у вигляді таблиці (див табл. 2.1).

Таблиця 2.1

Функціональні вимоги до системи логування подій

№	Функціональна вимога	Опис	Значення для безпеки
1	Збір логів	Отримання даних із різних джерел: серверів, клієнтських пристроїв, мережевого обладнання, систем безпеки	Забезпечує повноту інформації про події
2	Централізація логів	Об’єднання логів у єдиній системі зберігання	Дозволяє аналізувати події комплексно
3	Нормалізація даних	Приведення логів до єдиного формату	Полегшує обробку та аналіз
4	Збереження логів	Надійне зберігання даних протягом визначеного часу	Забезпечує можливість подальшого аналізу
5	Резервне копіювання	Створення резервних копій журналів подій	Запобігає втраті даних
6	Захист логів	Обмеження доступу, контроль змін	Забезпечує цілісність і безпеку
7	Доступ до логів	Надання доступу користувачам відповідно до ролей	Контроль використання інформації
8	Пошук та фільтрація	Можливість швидкого пошуку подій	Підвищує ефективність аналізу
9	Формування звітів	Генерація аналітичних звітів	Підтримує аудит і контроль

Продовження таблиці 2.1

№	Функціональна вимога	Опис	Значення для безпеки
10	Інтеграція з SIEM	Передача логів у системи аналізу	Забезпечує виявлення інцидентів

Узагальнюючи наведені функціональні вимоги, можна зробити висновок, що ефективна система логування повинна забезпечувати не лише збір і зберігання даних, але й створювати умови для їх безпечного використання, аналізу та інтеграції з іншими компонентами системи кібербезпеки організації.

Окремо слід врахувати, що вимоги до логування подій формуються не довільно, а базуються на міжнародних стандартах і практиках у сфері інформаційної безпеки. Зокрема, стандарт ISO/IEC 27001 визначає необхідність ведення журналів подій як одного з механізмів забезпечення контролю доступу та підзвітності дій у системі. У межах цього стандарту наголошується на обов’язковості фіксації подій безпеки, збереженні логів протягом визначеного часу та забезпеченні їх захисту від несанкціонованих змін [10].

Подібний підхід використовується і в рекомендаціях NIST, де журналювання розглядається як один із ключових інструментів моніторингу та реагування на інциденти. Зокрема, у документах NIST визначається необхідність централізованого збору логів, синхронізації часу між системами, а також можливості подальшого аналізу подій для виявлення аномальної активності. Також підкреслюється важливість забезпечення доступності логів для уповноважених осіб при одночасному обмеженні несанкціонованого доступу.

На практичному рівні організація процесу логування ґрунтується на ряді базових принципів. Передусім це повнота фіксації подій, яка передбачає запис усіх дій, що мають значення для безпеки системи. Не менш важливою є точність даних, яка забезпечується коректною синхронізацією часу та правильним визначенням джерела події. Крім того, логування повинно бути безперервним, щоб уникнути втрати інформації, а також захищеним — із застосуванням механізмів контролю доступу та захисту від змін. Окрему увагу приділяють

можливості подальшого аналізу логів, що передбачає їх структурованість і сумісність із системами моніторингу.

Окрім функціональних вимог, при проєктуванні системи логування необхідно враховувати нефункціональні вимоги, які визначають якість, надійність та ефективність її роботи. Вони не описують конкретні дії системи, але задають умови, за яких ці дії повинні виконуватись.

Однією з основних вимог є продуктивність системи. Логування не повинно суттєво впливати на роботу основних сервісів, навіть за умов великого обсягу подій. Система має забезпечувати швидкий запис і обробку логів, а також їх оперативну передачу до централізованого сховища без затримок.

Не менш важливою є вимога безпеки. Журнали подій містять чутливу інформацію, тому доступ до них повинен бути обмежений, а самі дані — захищені від несанкціонованого перегляду або змін. Для цього застосовуються механізми аутентифікації, розмежування прав доступу та криптографічного захисту [11].

Також важливою є масштабованість системи. У процесі розвитку інформаційної інфраструктури обсяг логів постійно зростає, тому система повинна підтримувати можливість розширення без втрати продуктивності. Це стосується як зберігання даних, так і їх обробки.

Окрему увагу слід приділити надійності та відмовостійкості. Система логування повинна працювати безперервно, навіть у разі збоїв окремих компонентів. Для цього передбачаються механізми резервування, дублювання даних та автоматичного відновлення. Крім того, важливою є зручність використання. Система повинна забезпечувати зрозумілий інтерфейс для роботи з логами, можливість швидкого пошуку, фільтрації та аналізу подій, що значно спрощує роботу фахівців з кібербезпеки.

Для узагальнення нефункціональних вимог до системи логування доцільно подати їх у вигляді таблиці (див табл. 2.2).

Таблиця 2.2

Нефункціональні вимоги до системи логування

№	Вимога	Опис	Значення
1	Продуктивність	Швидка обробка та запис логів	Не впливає на роботу системи
2	Безпека	Захист даних від несанкціонованого доступу	Забезпечує цілісність і конфіденційність
3	Масштабованість	Можливість розширення системи	Підтримує зростання обсягів даних
4	Надійність	Стабільна робота системи	Запобігає втраті логів
5	Відмовостійкість	Робота при збоях	Забезпечує безперервність
6	Доступність	Швидкий доступ до логів	Підвищує ефективність аналізу
7	Зручність використання	Інтерфейс та інструменти роботи	Полегшує роботу аналітиків

Урахування нефункціональних вимог дозволяє забезпечити стабільну, безпечну та ефективну роботу системи логування, що є необхідною умовою для подальшого аналізу подій і формування достовірної доказової бази.

2.2 Побудова архітектури системи збору, зберігання та захисту логів

У даному підрозділі розглядаються питання побудови архітектури системи збору, зберігання та захисту журналів подій у системах кібербезпеки організації. Основну увагу приділено структурі системи, взаємодії її компонентів та механізмам забезпечення цілісності й захищеності логів на всіх етапах їх обробки.

У межах підрозділу буде описано принципи організації процесу збору подій із різних джерел інформаційної інфраструктури, зокрема серверів, мережевого обладнання, операційних систем, засобів автентифікації та прикладного програмного забезпечення. Також буде розглянуто підходи до централізованого зберігання логів, використання баз даних і механізмів резервного копіювання для забезпечення надійності збереження інформації.

2.2.1 Моделі організації збору та централізації логів

Організація збору журналів подій є базовим елементом архітектури системи логування, оскільки саме на цьому етапі формується інформаційна основа для подальшого зберігання, аналізу та забезпечення цілісності даних. У сучасних інформаційних системах використовується декілька підходів до збору логів, які можуть застосовуватися окремо або в комбінації залежно від структури інфраструктури та вимог до безпеки.

Використання агентів – це один із підходів, які встановлюються безпосередньо на вузлах системи — серверах, робочих станціях або інших пристроях. Такі агенти виконують збір логів із локальних джерел, попередню обробку даних і їх передачу до централізованого сховища. Перевагою цього підходу є гнучкість і можливість адаптації до різних форматів даних, а також забезпечення безперервного збору інформації навіть у складних розподілених середовищах. Водночас використання агентів потребує додаткових ресурсів і налаштування, що може ускладнювати їх впровадження у великих системах [12].

Альтернативним підходом є використання протоколу syslog, який є стандартом для передачі журналів подій у мережевих середовищах. У цьому випадку пристрої та програмні системи надсилають повідомлення про події на спеціалізований сервер логування без необхідності встановлення додаткового програмного забезпечення. Такий підхід відзначається простотою реалізації та широкою підтримкою з боку різних систем і мережевого обладнання. Разом із тим він має обмеження, пов’язані з безпекою передачі даних, оскільки базові реалізації syslog не передбачають шифрування, що потребує використання додаткових механізмів захисту.

У системах також широко застосовується інтеграція через програмні інтерфейси (API), які дозволяють отримувати дані про події безпосередньо з програмних сервісів, хмарних платформ або веб-застосунків. Використання API забезпечує високий рівень гнучкості та дозволяє працювати з різноманітними

джерелами даних, включаючи ті, що не підтримують стандартні протоколи логування. Такий підхід особливо актуальний у хмарних і мікросервісних архітектурах, де традиційні методи збору логів можуть бути недостатніми.

Усі зазначені підходи, як правило, поєднуються в межах централізованих систем збору та обробки логів, зокрема SIEM-рішень. Централізація логів дозволяє об’єднати дані з різних джерел в єдиному середовищі, що спрощує їх аналіз, забезпечує кореляцію подій та підвищує ефективність виявлення інцидентів. Крім того, централізовані системи забезпечують можливість застосування політик безпеки, контролю доступу та механізмів захисту даних, що є важливим для забезпечення цілісності та достовірності журналів подій [13].

Таким чином, вибір моделі збору та централізації логів залежить від особливостей інформаційної інфраструктури організації, вимог до безпеки та необхідного рівня контролю над даними. Найбільш ефективним є комбінований підхід, який поєднує використання агентів, стандартних протоколів і API з централізованими системами обробки, що дозволяє забезпечити повноту збору інформації та її подальше використання в системах кібербезпеки.

Для кращого розуміння процесу збору та централізації журналів подій доцільно подати узагальнену схему архітектури системи логування. Вона відображає основні джерела формування логів, способи їх збору за допомогою агентів, протоколу syslog та API, а також подальшу передачу даних до централізованої SIEM-системи для обробки та аналізу. Така структура дозволяє побачити взаємозв’язок між компонентами системи та загальний потік даних у процесі логування (рис. 2.1).

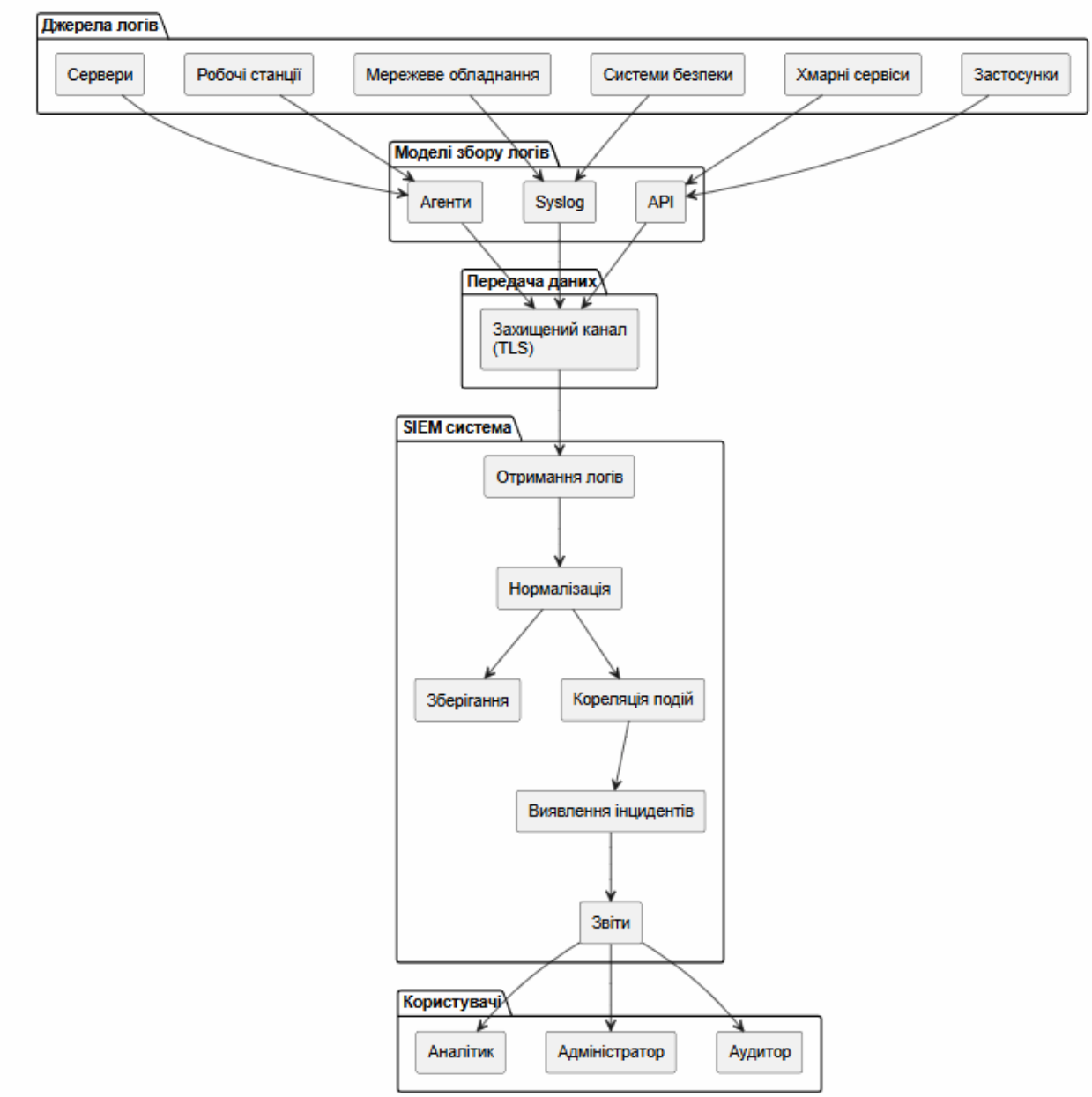

Рис. 2.1 Архітектура збору та централізації логів (вертикальна модель)

2.2.2 Засоби зберігання та захисту журналів подій

Після збору та централізації журналів подій наступним етапом є організація їх надійного зберігання та захисту. Сам факт отримання логів із різних джерел ще не гарантує їх практичної цінності для системи кібербезпеки. Щоб журнали подій могли використовуватись для моніторингу, аудиту, розслідування інцидентів і формування доказової бази, вони мають зберігатися у структурованому, доступному та захищеному середовищі. При цьому важливо

забезпечити не лише фізичне збереження даних, а й захист від несанкціонованого доступу, випадкового видалення, пошкодження або підробки.

Одним із засобів зберігання журналів подій є бази даних, вони дають змогу впорядковувати великі обсяги логів, виконувати пошук, фільтрацію, сортування та подальший аналіз подій за різними параметрами. У системах кібербезпеки можуть використовуватися як реляційні бази даних, так і спеціалізовані сховища, орієнтовані на роботу з великими потоками подій. Наприклад, реляційні бази доцільні для структурованого зберігання інформації про користувачів, події, час доступу, IP-адреси та результати перевірок. Водночас для обробки великої кількості логів частіше застосовуються індексовані сховища, які дозволяють швидко знаходити потрібні записи навіть за значного обсягу даних.

Під час використання баз даних необхідно враховувати, що журнали подій мають зберігатися не лише у зручному для аналізу вигляді, а й у формі, яка унеможливлює їх непомітну зміну. Для цього можуть застосовуватися механізми хешування записів, цифрового підпису, журналювання змін у самій базі даних, а також обмеження операцій редагування та видалення. Такий підхід дозволяє контролювати стан логів і виявляти спроби втручання в їхній зміст [14]. Окремо слід передбачити збереження службової інформації про кожен запис: дату і час створення, джерело події, ідентифікатор користувача або системного процесу, рівень критичності та результат перевірки цілісності.

Поряд із локальними базами даних у сучасних організаціях часто використовуються хмарні сховища. Вони дозволяють зберігати значні обсяги журналів подій без необхідності розгортання великої власної інфраструктури. Перевагою хмарного підходу є масштабованість, оскільки обсяг сховища може збільшуватися залежно від кількості логів. Крім того, хмарні сервіси зазвичай надають інструменти резервного копіювання, контролю доступу, шифрування та довгострокового архівного зберігання. Це особливо корисно для організацій, які мають розподілену інфраструктуру або використовують хмарні сервіси у своїй діяльності.

Разом із тим використання хмарних сховищ потребує уважного підходу до питань безпеки. Журнали подій можуть містити службову, конфіденційну або персональну інформацію, тому передавання та зберігання таких даних у хмарному середовищі повинно супроводжуватися шифруванням. Доцільно застосовувати шифрування під час передавання даних і під час їх зберігання. Також необхідно визначити політики доступу, правила зберігання, строки архівації та процедури видалення логів після завершення встановленого періоду. У випадку використання хмарної інфраструктури важливо контролювати, хто має доступ до журналів подій, які операції може виконувати та чи фіксуються ці дії в окремому журналі аудиту.

Контроль доступу є одним із основних засобів захисту журналів подій незалежно від того, де саме вони зберігаються — у локальній базі даних, SIEM-системі чи хмарному сховищі. Доступ до логів повинен надаватися лише тим користувачам, які мають відповідні повноваження. Для цього доцільно використовувати рольову модель доступу, за якої права користувачів визначаються відповідно до їхніх службових обов’язків. Наприклад, аналітик безпеки може мати право переглядати та аналізувати журнали подій, адміністратор — налаштовувати параметри збору й зберігання, а аудитор — отримувати доступ до звітів без можливості змінювати записи.

Принцип мінімальних привілеїв, відповідно до якого користувач отримує лише ті права, які необхідні для виконання його завдань. Такий підхід зменшує ризик несанкціонованих змін або випадкового пошкодження даних. Крім того, усі дії з журналами подій мають фіксуватися: перегляд, експорт, зміна налаштувань доступу, видалення або спроби виконання заборонених операцій. Це дозволяє створити додатковий рівень контролю та забезпечити підзвітність осіб, які працюють із доказовою інформацією.

Важливим елементом захисту журналів подій є резервування. Навіть якщо система має належні механізми контролю доступу та захисту від змін, залишається ризик втрати даних унаслідок технічного збою, помилки адміністратора, кібератаки або пошкодження основного сховища. Тому журнали

подій повинні регулярно копіюватися до резервного середовища. Резервні копії можуть зберігатися на окремих серверах, у хмарному архіві або на спеціалізованих носіях, які не доступні для звичайних користувачів системи.

Резервування має бути організоване таким чином, щоб забезпечити можливість відновлення логів за потрібний період часу. Для цього визначаються періодичність створення копій, строки їх зберігання та порядок перевірки працездатності резервів. Недостатньо лише створювати копії — необхідно періодично перевіряти, чи можна відновити дані з резервного сховища без втрат і пошкоджень. У контексті доказової бази важливо також контролювати цілісність резервних копій, наприклад шляхом збереження контрольних хеш-значень або використання цифрового підпису.

Зберігання журналів подій доцільно організовувати за принципом багаторівневого підходу. Поточні логи, які часто використовуються для аналізу, можуть зберігатися у швидкому сховищі з можливістю оперативного пошуку. Старіші записи, що потрібні переважно для аудиту або розслідувань, можуть переноситися до архівного сховища. Резервні копії при цьому мають зберігатися окремо від основної системи, щоб у разі пошкодження або компрометації основного середовища організація могла відновити необхідні дані.

Для наочного відображення процесу зберігання та захисту журналів подій доцільно подати узагальнену архітектурну схему, яка демонструє основні компоненти системи та їх взаємодію. На схемі показано розподіл логів між різними рівнями зберігання, використання баз даних і хмарних сховищ, а також застосування механізмів контролю доступу, шифрування та резервування. Це дозволяє зрозуміти, як забезпечується збереження, захист і доступність журналів подій у системах кібербезпеки організації (рис. 2.2).

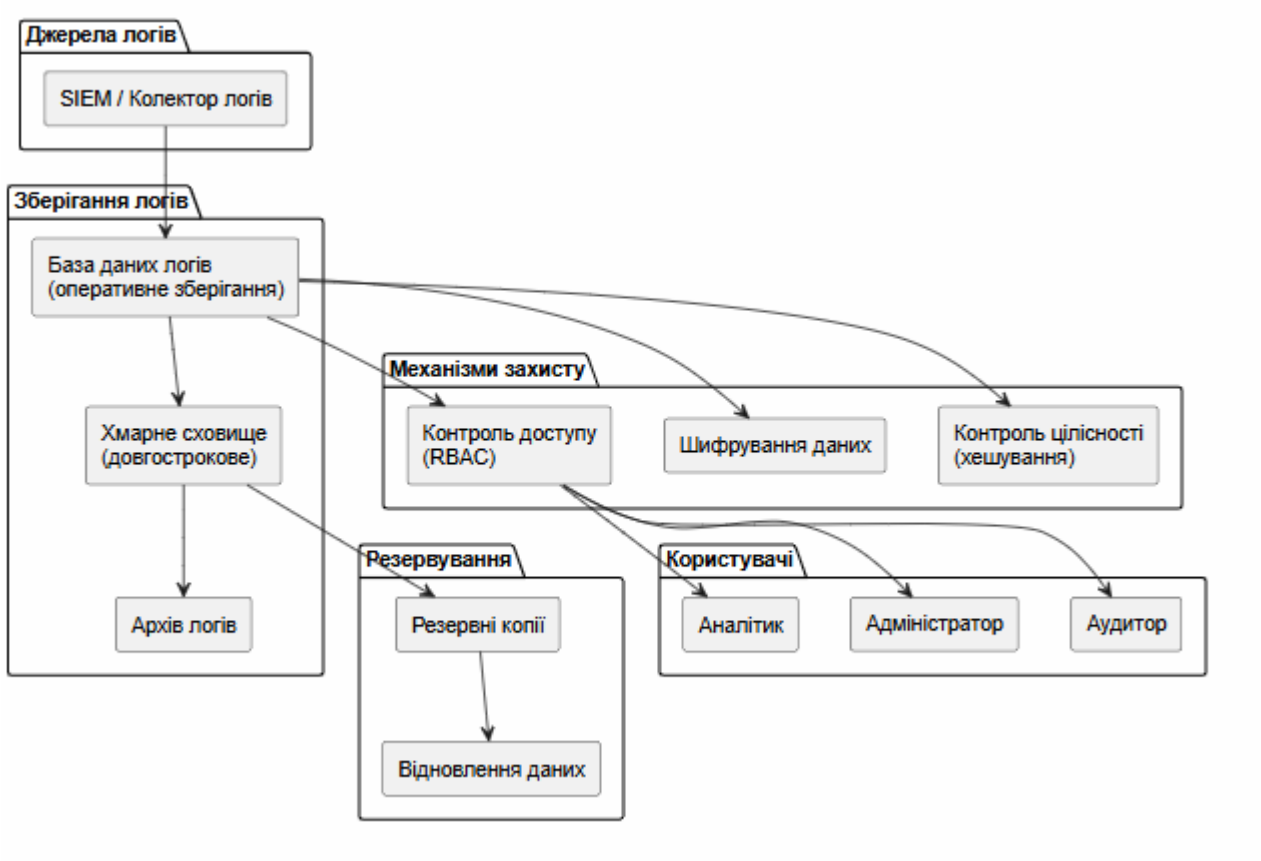

Рис. 2.2 Архітектура зберігання та захисту журналів подій

Засоби зберігання та захисту журналів подій повинні розглядатися як єдина система, що поєднує бази даних, хмарні сховища, контроль доступу та резервування. Їхнє правильне поєднання дозволяє забезпечити доступність логів для аналізу, захистити їх від несанкціонованих змін і зберегти придатність журналів подій для використання як доказової бази у процесах кібербезпеки організації.

2.3 Реалізація можливостей механізмів забезпечення цілісності та захисту даних

У даному підрозділі розглядаються практичні аспекти реалізації механізмів забезпечення цілісності та захисту даних у системах кібербезпеки організації. Основну увагу приділено програмним і криптографічним засобам, що дозволяють запобігати несанкціонованій зміні, видаленню або підробці

журналів подій та іншої критично важливої інформації.

У межах підрозділу буде описано реалізацію механізмів контролю цілісності даних із використанням хешування, цифрових підписів та засобів шифрування. Також буде розглянуто підходи до перевірки достовірності журналів подій, організації контролю доступу до даних і використання механізмів аудиту для фіксації дій користувачів та системних процесів.

2.3.1 Криптографічні методи забезпечення цілісності (хешування, цифровий підпис)

Забезпечення цілісності журналів подій є однією з основних умов їх подальшого використання як достовірної доказової бази. Якщо записи в логах можуть бути непомітно змінені, видалені або підроблені, то вони втрачають цінність для розслідування інцидентів інформаційної безпеки. Саме тому під час проєктування системи захисту журналів подій необхідно передбачити механізми, які дозволяють перевірити, чи залишилися дані незмінними з моменту їх створення або збереження.

Одним із найпоширеніших способів перевірки цілісності є хешування. Хешування — це процес перетворення вхідних даних будь-якого розміру у фіксоване значення, яке називається хешем. Для журналів подій такий підхід є зручним, оскільки навіть незначна зміна в записі призводить до повністю іншого хеш-значення. Наприклад, якщо у журналі подій змінити лише одну цифру в часі входу користувача або IP-адресі, результат хешування вже не збігатиметься з початковим значенням. Це дозволяє швидко виявляти факт втручання в дані.

У системах кібербезпеки для таких завдань часто використовується алгоритм SHA-256. Він формує хеш довжиною 256 біт і застосовується для перевірки незмінності даних, зокрема файлів, повідомлень, записів у базах даних та журналів подій. У контексті логування SHA-256 може застосовуватися до окремого запису, групи записів або цілого файлу журналу. Наприклад, після створення запису про невдалу спробу входу в систему для нього обчислюється

хеш, який зберігається окремо [15]. Під час подальшої перевірки система повторно обчислює хеш цього запису та порівнює його з початковим. Якщо значення збігаються, запис вважається незміненим; якщо ні — це свідчить про можливе порушення цілісності.

Практичне використання хешування можна розглянути на прикладі запису журналу подій:

2026-04-12 10:25:33 | user_admin | login_failed | IP: 192.168.1.45

Для цього запису система обчислює хеш-значення та зберігає його в окремому полі або таблиці. Якщо зловмисник змінить подію, наприклад із login_failed на login_success, то новий хеш уже не відповідатиме збереженому. Таким чином, навіть якщо зміну важко помітити візуально, вона буде виявлена під час криптографічної перевірки.

Разом із тим хешування саме по собі не завжди достатнє. Якщо зловмисник має доступ не лише до логів, а й до збережених хешів, він може змінити запис і заново обчислити для нього нове хеш-значення. Тому для підвищення рівня захисту доцільно використовувати цифровий підпис. Цифровий підпис дозволяє не лише перевірити цілісність даних, а й підтвердити їх походження. Це означає, що система або відповідальний компонент, який сформував журнал подій, може підписати запис за допомогою приватного ключа, а перевірка здійснюється за допомогою відкритого ключа.

У випадку журналів подій цифровий підпис може застосовуватися до окремих записів, пакетів логів або архівів. Наприклад, SIEM-система отримує події за певний проміжок часу, формує блок записів, обчислює для нього хеш, а потім підписує цей хеш приватним ключем. Надалі будь-яка уповноважена особа може перевірити підпис і переконатися, що дані не були змінені після підписання. Це особливо корисно у випадках, коли журнали подій використовуються під час аудиту або розслідування інцидентів, оскільки дозволяє підтвердити автентичність доказової інформації.

Наприклад, після завершення робочого дня система може сформувати архів логів за добу, обчислити для нього хеш SHA-256 і накласти цифровий

підпис. У разі розслідування інциденту цей архів можна перевірити: якщо підпис є коректним, це підтверджує, що з моменту підписання дані не були змінені. Якщо перевірка не проходить, це свідчить про можливе пошкодження або несанкціоноване втручання.

Ще одним підходом є використання ланцюжків хешів. Такий механізм передбачає, що кожен новий запис журналу містить не лише власний хеш, а й хеш попереднього запису. У результаті формується послідовність, у якій кожен елемент пов’язаний із попереднім. Якщо змінити один запис у середині журналу, порушиться весь подальший ланцюжок, оскільки хеші наступних записів перестануть відповідати один одному.

Цей підхід схожий на принцип роботи blockchain, де блоки даних пов’язані між собою за допомогою хешів. Проте для системи логування не обов’язково створювати повноцінний блокчейн. Достатньо використати саму ідею послідовного зв’язування записів, щоб ускладнити непомітну зміну журналу подій. Наприклад, кожен запис може містити такі поля: час події, користувач, дія, хеш поточного запису та хеш попереднього запису. Якщо зловмисник змінить старий запис, йому доведеться перерахувати всі наступні хеші, що значно ускладнює приховане втручання.

Приклад логічної структури такого запису може виглядати так:

$$Log\_3 = SHA\text{-}256(\text{дані події } 3 + hash(Log\_2))$$

У цьому випадку третій запис залежить від другого, четвертий — від третього, і так далі. Завдяки цьому журнал подій набуває властивості послідовної перевірки, коли порушення одного елемента одразу впливає на цілісність усієї наступної частини журналу.

Для узагальнення криптографічних методів забезпечення цілісності журналів подій доцільно подати їх у вигляді таблиці (див табл. 2.3).

Таблиця 2.3

Криптографічні методи забезпечення цілісності журналів подій

№	Метод	Сутність методу	Приклад застосування	Результат
1	Хешування SHA-256	Обчислення унікального хеш-значення для запису або файлу логів	Формування хешу для кожного запису журналу подій	Виявлення змін у логах
2	Цифровий підпис	Підписання хешу або блоку логів приватним ключем	Підпис добового архіву журналів подій	Підтвердження цілісності та походження даних
3	Ланцюжок хешів	Зв’язування кожного запису з попереднім через хеш	Додавання до нового запису хешу попереднього запису	Виявлення зміни будь-якого запису в послідовності
4	Хешування архівів	Обчислення хешу для групи логів за певний період	Хешування журналів за день або тиждень	Контроль незмінності архівних даних
5	Комбінований підхід	Поєднання хешування, підпису та контролю доступу	Хешування записів, підпис архівів, обмеження прав доступу	Підвищення надійності захисту доказової бази

Застосування криптографічних методів дозволяє не лише виявляти факт зміни журналів подій, а й підвищувати довіру до даних, які використовуються під час аудиту або розслідування інцидентів. Хешування забезпечує швидку перевірку незмінності записів, цифровий підпис підтверджує походження та автентичність даних, а ланцюжки хешів дозволяють контролювати послідовність подій у журналі. У поєднанні ці механізми формують основу для захисту логів і забезпечення їх придатності як доказової бази в системах кібербезпеки організації.

2.3.2 Механізми контролю змін та аудиту журналів подій

Криптографічні методи дозволяють виявити факт зміни даних, однак для повноцінного захисту журналів подій необхідні організаційні та програмні механізми, які забезпечують контроль доступу, фіксацію всіх дій із логами та своєчасне реагування на підозрілу активність. Такі механізми формують додатковий рівень захисту і дозволяють не лише визначити, що дані були

змінені, але й встановити, ким, коли і за яких умов це відбулося.

Одним із ключових елементів є аудит журналів подій. Аудит передбачає регулярну перевірку логів і дій користувачів, пов’язаних із їх переглядом, зміною або видаленням. У межах аудиту здійснюється аналіз відповідності дій встановленим політикам безпеки, виявлення аномалій та фіксація всіх операцій над журналами. Наприклад, якщо адміністратор виконує масове видалення записів або змінює параметри зберігання логів, така дія повинна бути автоматично зафіксована та доступна для перевірки. У сучасних системах аудит часто реалізується як окремий рівень журналювання, де записуються всі дії, пов’язані з управлінням логами, що дозволяє створити прозору систему контролю.

Важливим механізмом є також контроль версій журналів подій. Цей підхід передбачає збереження різних станів логів або їхніх змін у вигляді послідовних версій. У разі внесення змін система зберігає попередню версію даних, що дозволяє відновити їхній початковий стан. Наприклад, якщо певний запис був змінений або видалений, контроль версій дозволяє повернутись до попереднього варіанту та визначити, які саме зміни були внесені. Такий підхід особливо корисний у випадках внутрішніх розслідувань, коли необхідно відстежити історію змін даних.

Ще одним елементом є журнал змін, який фіксує всі операції над журналами подій. На відміну від основних логів, які відображають події системи, журнал змін містить інформацію про дії користувачів або процесів щодо самих логів. До таких дій належать перегляд, експорт, редагування, видалення або зміна параметрів доступу. Наприклад, запис у журналі змін може містити такі дані: ідентифікатор користувача, тип операції, час виконання, об’єкт змін і результат дії. Це дозволяє створити повну картину взаємодії з журналами подій і забезпечити можливість контролю навіть у випадках спроб приховування слідів.

Для оперативного реагування на підозрілу активність використовуються alert-системи, які автоматично генерують повідомлення у разі виявлення

небезпечних або аномальних дій. Такі системи можуть бути інтегровані з SIEM-рішеннями або працювати як окремий компонент. Наприклад, сигнал може формуватися у випадках несанкціонованого доступу до логів, спроби їх масового видалення, зміни прав доступу або порушення цілісності даних. Повідомлення можуть надсилатися адміністраторам у вигляді електронних листів, повідомлень у месенджерах або відображатися в системі моніторингу. Це дозволяє швидко реагувати на інциденти та запобігати їх подальшому розвитку.

Практичне застосування таких механізмів можна розглянути на прикладі. Якщо користувач із правами адміністратора намагається змінити записи журналу подій, система фіксує цю дію в журналі змін, зберігає попередню версію даних, а також генерує alert-повідомлення для служби безпеки. Одночасно може бути запущена процедура перевірки цілісності логів за допомогою хешування. У разі виявлення невідповідності система може обмежити доступ до журналів або заблокувати обліковий запис користувача до з’ясування обставин (див табл. 2.4).

Таблиця 2.4

Механізми контролю змін та аудиту журналів подій

№	Механізм	Сутність	Приклад застосування	Результат
1	Аудит	Перевірка дій користувачів і системи щодо логів	Аналіз доступу до журналів подій	Виявлення порушень політик
2	Контроль версій	Збереження попередніх станів даних	Відновлення зміненого запису	Можливість відстеження змін
3	Журнал змін	Фіксація всіх операцій над логами	Запис дій адміністратора	Повний контроль доступу
4	Alert-системи	Генерація повідомлень про підозрілі дії	Повідомлення про видалення логів	Оперативне реагування
5	Комбінований підхід	Поєднання всіх механізмів	Аудит + alerts + контроль версій	Підвищення рівня захисту

Застосування механізмів аудиту, контролю версій, журналів змін та систем сповіщення дозволяє створити комплексну систему контролю над журналами подій. У поєднанні з криптографічними методами це забезпечує не лише виявлення змін, а й повне відстеження дій із даними, що є критично важливим

для формування достовірної доказової бази в системах кібербезпеки організації.

2.4 Методичні підходи щодо забезпечення цілісності та перевірки достовірності доказової бази

У процесі реагування на кіберінциденти доказова база має цінність лише за умови, що її цілісність підтверджена, а джерело та шлях отримання даних є прозорими й відтворюваними. Тому методичний підхід до роботи з цифровими доказами повинен поєднувати технічні механізми контролю (FIM, логування, хешування) з процедурними вимогами (ланцюг збереження, журнал дій аналітика, відтворюваність аналізу).

Практична цінність платформи Wazuh у цьому контексті полягає в тому, що вона дає змогу організувати безперервний моніторинг стану кінцевих точок і централізовано акумулювати події безпеки. Зокрема, моніторинг цілісності файлів (File Integrity Monitoring) дозволяє фіксувати критичні зміни у файловій системі: створення, модифікацію, видалення або зміну атрибутів файлів. Для доказового процесу це важливо, оскільки дає змогу встановити часову послідовність подій та прив’язати зміну до конкретного хоста, користувача або процесу.

Методика забезпечення цілісності доцільно реалізується у кілька етапів. На першому етапі визначається перелік критичних об’єктів контролю: системні каталоги, конфігураційні файли, журнали автентифікації, службові скрипти, ключі доступу. На другому етапі налаштовується агентний збір подій із нормалізованою структурою журналів, щоб дані з різних ОС могли бути порівняні в єдиному форматі. На третьому етапі впроваджується хеш-контроль (наприклад, SHA-256) для незалежної перевірки незмінності копій доказових матеріалів під час передачі та аналізу.

Окремим елементом методики є перевірка достовірності даних через кореляцію подій із різних джерел. Наприклад, зміна критичного файлу на кінцевій точці повинна зіставлятися з записами автентифікації, мережевою

активністю та адміністративними діями. Якщо в часовому проміжку, де зафіксовано модифікацію, присутні підозрілі входи або нетипові мережеві з'єднання, це підсилює доказову вагу інциденту. Таким чином, достовірність формується не лише окремим логом, а узгодженістю всієї сукупності телеметрії.

Щоб мінімізувати ризик спотворення висновків, доцільно дотримуватись принципу відтворюваності: усі дії аналітика мають бути формалізовані як повторюваний сценарій. У протоколі фіксуються час отримання даних, версії інструментів, параметри фільтрації, правила кореляції та підсумкові індикатори компрометації. Це підвищує об'єктивність експертизи та дозволяє повторно верифікувати результати незалежною стороною [16, 17].

Підсумовуючи, методичні підходи до забезпечення цілісності та перевірки достовірності доказової бази мають будуватися на поєднанні:

1. безперервного FIM-моніторингу;
2. централізованого збору та нормалізації логів;
3. криптографічного хеш-контролю;
4. кореляційного аналізу подій;
5. документованого ланцюга збереження доказів.

Саме така комбінація дає підстави вважати цифрові докази технічно надійними та процесуально обґрунтованими.

На рис. 2.3 наведено приклад обчислення контрольної суми SHA-256 для підтвердження незмінності цифрового доказу.

```python
import hashlib
from pathlib import Path

def sha256_file(path: str) -> str:
    h = hashlib.sha256()
    with open(path, "rb") as f:
        for chunk in iter(lambda: f.read(8192), b""):
            h.update(chunk)
    return h.hexdigest()

evidence = "disk_image.E01"
print("SHA-256:", sha256_file(evidence))
```

Рис. 2.3 Обчислення SHA-256 для перевірки цілісності доказового файлу

Отримане значення гешу використовується як еталон під час усіх подальших перевірок, передач і експертного аналізу.

На рис. 2.4 показано приклад базового сценарію контролю змін у критичному каталозі ОС Linux.

```
#!/bin/bash
TARGET="/etc"
BASELINE="/var/tmp/etc_baseline.sha256"
CURRENT="/var/tmp/etc_current.sha256"

find "$TARGET" -type f -exec sha256sum {} \; | sort > "$CURRENT"

if [ ! -f "$BASELINE" ]; then
  cp "$CURRENT" "$BASELINE"
  echo "Baseline created."
  exit 0
fi

if diff -u "$BASELINE" "$CURRENT" > /var/tmp/etc_diff.txt; then
  echo "No integrity changes detected."
else
  echo "WARNING: Integrity changes found!"
fi
```

Рис. 2.4 Скрипт первинного контролю цілісності системних файлів

Цей програмний скрипт автоматизує процес моніторингу цілісності, обчислюючи криптографічні хеш-суми SHA-256 для всіх файлів у заданій системній директорії. Під час свого першого запуску сценарій інвентаризує каталог і формує еталонний стан системи (baseline), зберігаючи ці показники у спеціальний файл. При всіх подальших виконаннях алгоритм генерує нові контрольні суми та автоматично зіставляє їх із первинною базою за допомогою системної утиліти порівняння. Таким чином, регулярне порівняння еталонного та поточного стану файлової системи дозволяє швидко виявити несанкціоновані модифікації, пошкодження критичних конфігурацій або приховані сліди втручання зловмисників.

На рис. 2.5 представлено UML-модель методики перевірки цілісності та достовірності доказової бази [18-20].

Рис. 2.5 UML-діаграма процесу верифікації цифрових доказів

Діаграма відображає послідовність дій від налаштування контролю до підготовки обґрунтованого експертного висновку.

Висновок до розділу 2

У другому розділі було досліджено теоретико-методичні засади побудови процесу реагування на кіберінциденти та визначено роль сучасних платформ моніторингу в цьому процесі. Розглянуто підходи до виявлення загроз, обробки подій безпеки, оцінювання ризиків і формування сценаріїв реагування, що дало змогу обґрунтувати вимоги до структури майбутньої системи захисту журналів подій.

Окрему увагу приділено методичним підходам до забезпечення цілісності та перевірки достовірності доказової бази. Визначено, що надійність цифрових доказів досягається через поєднання безперервного моніторингу змін, централізованого логування, криптографічного контролю цілісності, кореляційного аналізу та документування ланцюга збереження доказів. Така сукупність заходів створює підґрунтя для об’єктивного, відтворюваного й процесуально коректного розслідування інцидентів.

Узагальнення результатів другого розділу дозволило сформувати концептуальну основу практичної частини роботи. Отримані висновки стали методичною базою для подальшої реалізації архітектури збору журналів, впровадження механізмів захисту даних і побудови процесу формування доказової бази в третьому розділі. Отже, завдання аналітичного етапу виконано, а сформовані положення забезпечили логічний перехід від теорії до прикладної реалізації системи.

РОЗДІЛ 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ЖУРНАЛІВ ПОДІЙ

3.1 Реалізація підсистеми збору та збереження журналів подій

Підсистема збору та збереження журналів подій є базовим елементом системи реагування на інциденти, оскільки саме вона формує первинну інформаційну основу для виявлення атак, аналізу аномалій та підготовки доказової бази. У межах практичної реалізації було побудовано централізовану архітектуру збору логів із кінцевих точок і мережевих джерел із подальшим збереженням у сховищі, придатному для швидкого пошуку та кореляції подій.

Функціонально підсистема реалізує такі завдання:

1. безперервний збір журналів із хостів Windows/Linux;
2. приймання подій від засобів мережевого моніторингу;
3. нормалізацію подій до уніфікованого формату;
4. централізоване збереження та індексацію;
5. підтримку цілісності і часової узгодженості записів [21, 22].

У практичній частині як центральний вузол збору використано сервер Wazuh, до якого підключено агентів кінцевих точок. На хості Linux було активовано моніторинг системних журналів автентифікації, подій доступу та змін у файловій системі, а на Windows-станції налаштовано збір подій входу/виходу, помилок безпеки та системних сповіщень з Event Viewer. Для розширення видимості мережевого рівня інтегровано події IDS/мережевого сенсора, що дозволило фіксувати сканування, підозрілі з’єднання та нетипову активність у трафіку.

Після надходження даних виконується їх первинна обробка: перевірка структури події, присвоєння міток джерела, рівня критичності та часових атрибутів. Нормалізовані події зберігаються в централізованому індексованому сховищі, що забезпечує швидке виконання запитів і побудову часових ланцюгів інциденту. Такий підхід дозволяє не лише здійснювати оперативний моніторинг,

а й проводити ретроспективний аналіз після завершення інциденту.

Для забезпечення надійності збереження журналів реалізовано організаційно-технічні обмеження: розмежування прав доступу до логів, фіксацію службових дій адміністраторів, а також регламент резервного копіювання архівних індексів. Це знижує ризик втрати подій або несанкціонованого коригування історичних записів. Крім того, часову коректність журналів підтримано через синхронізацію системного часу вузлів, що критично важливо для правильної кореляції подій із різних джерел.

На рис. 3.1 представлено архітектуру системи збору, захисту та централізованого збереження журналів подій [23].

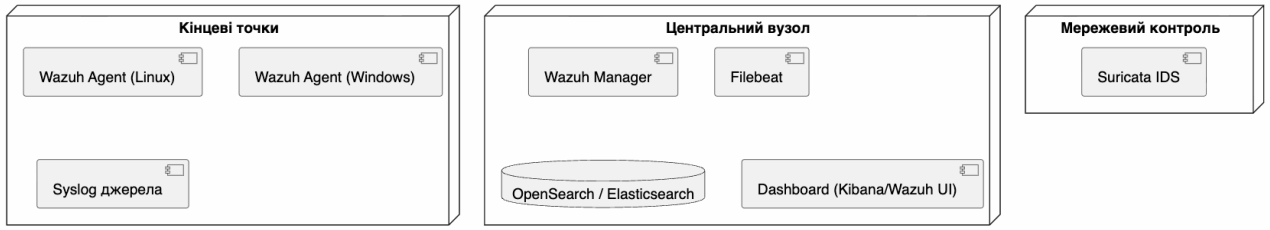

Рис. 3.1 Архітектура системи захисту журналів подій

Діаграма відображає повний маршрут даних: від кінцевих точок і мережевих сенсорів до індексації та подальшого аналітичного використання. Спочатку система отримує події від агента, приводить їх до нормального вигляду, додає необхідні дані та записує у сховище. Після цього для архіву логів створюється унікальний цифровий відбиток — хеш SHA-256. Далі цей відбиток перевіряють на справжність. Якщо він збігається з еталоном, система позначає запис як цілісний. Якщо ж не збігається, система створює тривожне сповіщення про порушення цілісності та передає цей інцидент до SOC на розслідування. Наприкінці, незалежно від того, як саме завершилась перевірка, відбувається ротація та обов'язкове резервне копіювання всіх даних для їх безпечного збереження.

Перевірка цілісності наведена на рис. 3.2.

Рис. 3.2 Процес перевірки цілісності журналів подій

Фрагмент конфігурації Wazuh Agent (Linux, FIM + syscheck)

```
<ossec_config>
  <syscheck>
    <disabled>no</disabled>
    <frequency>3600</frequency>
    <scan_on_start>yes</scan_on_start>
    <directories check_all="yes">/etc,/var/log</directories>
    <ignore>/var/log/wtmp</ignore>
    <ignore>/var/log/btmp</ignore>
  </syscheck>
</ossec_config>
```

Фрагмент rules (реакція на зміну критичних файлів)

```
<group name="custom_integrity,">
```

```
  <rule id="100501" level="10">
    <if_group>syscheck</if_group>
    <match>/etc/passwd|/etc/shadow|/etc/sudoers</match>
    <description>Критична зміна системного файлу</description>
    <mitre>
      <id>T1078</id>
    </mitre>
  </rule>
</group>
```

Скрипт перевірки архіву логів наведений на рис. 3.3.

```python
import hashlib
from pathlib import Path

LOG_ARCHIVE = "wazuh-archives-2026-05-06.tar.gz"
HASH_FILE = "wazuh-archives-2026-05-06.sha256"

def sha256(path):
    h = hashlib.sha256()
    with open(path, "rb") as f:
        for chunk in iter(lambda: f.read(8192), b""):
            h.update(chunk)
    return h.hexdigest()

digest = sha256(LOG_ARCHIVE)
Path(HASH_FILE).write_text(digest + "  " + LOG_ARCHIVE + "\n", encoding="utf-8")
print("Stored hash:", digest)
```

Рис. 3.3 Програмна верифікація цілісності архіву журналів

Для оцінки обсягів отриманої телеметрії було проаналізовано роботу підсистеми збору даних. Результати розподілу зафіксованих подій за джерелами протягом 24 годин наведено в (див табл. 3.1).

Таблиця 3.1

Розподіл подій за джерелами за 24 години

Джерело	Кількість подій	Частка, %	Критичні
Linux Agent	18 420	41.3	37
Windows Agent	20 115	45.1	52
Suricata IDS	4 730	10.6	29
Syslog (мережеві пристрої)	1 380	3.0	4

Продовження таблиці 3.1

Джерело	Кількість подій	Частка, %	Критичні
Разом	44 645	100	122

Отримані показники підтверджують домінування endpoint-телеметрії у загальному потоці журналів, що відповідає обраній архітектурі з агентним збором подій. Також варто звернути увагу на показники після реалізації (див табл. 3.2).

Таблиця 3.2

Порівняння показників до та після реалізації підсистеми

Показник	До реалізації	Після реалізації	Покращення
Середній час виявлення інциденту (MTTD), хв	47	12	-74,5%
Середній час реагування (MTTR), хв	95	34	-64,2%
Частка подій без атрибуції джерела	18%	2%	-16 п.п.
Події з підтвердженою цілісністю	61%	98%	+37 п.п

3.2 Реалізація механізмів забезпечення цілісності та захисту даних

У межах реалізації системи захисту журналів подій ключовий акцент зроблено на трьох практичних механізмах:

1. контроль цілісності файлів та журналів;
2. криптографічний захист даних під час збереження і передачі;
3. контроль доступу та аудит адміністративних дій [24, 25].

Технічно це реалізовано через поєднання Wazuh FIM, хеш-контролю архівів логів, TLS-захисту каналу передачі, рольової моделі доступу та

журналювання дій операторів. Такий підхід дозволяє одночасно виявляти несанкціоновані зміни, підтверджувати незмінність доказових даних і зменшувати ризик внутрішніх порушень.

На рис. 3.4 показано процес технічного підтвердження цілісності журналів подій.

Рис. 3.4 Процес забезпечення цілісності та верифікації журналів

У разі невідповідності контрольної суми ініціюється окремий сценарій реагування з фіксацією інциденту та передачею його в SOC. Важливим кроком цього сценарію, як показано на алгоритмі, є також блокування доступу до

зміненого набору даних. Це робиться для того, щоб ізолювати скомпрометовану інформацію та уникнути її використання в подальшій аналітиці або під час розслідувань.

Натомість, якщо повторна перевірка підтверджує збіг обчисленого хешу з еталонним, система автоматично маркує, що дані підтверджені як цілісні. Це гарантує їхню достовірність і юридичну значущість для майбутнього аудиту.

На рис. 3.5 показано фрагмент реалізації 1: контроль цілісності критичних файлів (Wazuh)

```
<ossec_config>
  <syscheck>
    <disabled>no</disabled>
    <scan_on_start>yes</scan_on_start>
    <frequency>1800</frequency>
    <directories check_all="yes">/etc,/var/log,/opt/security</directories>
    <alert_new_files>yes</alert_new_files>
    <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>
  </syscheck>
</ossec_config>
```

Рис. 3.5 Налаштування FIM для контролю критичних директорій

На рис. 3.6 показано фрагмент реалізації 2: перевірка цілісності архіву логів

```
#!/bin/bash
ARCHIVE="events-archive-2026-05-06.tar.gz"
HASH_FILE="${ARCHIVE}.sha256"

sha256sum "$ARCHIVE" > "$HASH_FILE"
echo "Baseline hash saved: $HASH_FILE"

# Перевірка під час подальшого використання
sha256sum -c "$HASH_FILE"
```

Рис. 3.6 Формування та перевірка контрольної суми архіву журналів

На рис. 3.7 показано фрагмент реалізації 3: шифрування резервної копії

```
#!/bin/bash
INPUT="events-archive-2026-05-06.tar.gz"
OUTPUT="${INPUT}.enc"

openssl enc -aes-256-cbc -pbkdf2 -salt -in "$INPUT" -out "$OUTPUT"
echo "Encrypted backup created: $OUTPUT"
```

Рис. 3.7 Шифрування резервної копії журналів алгоритмом AES-256

Для оцінки ефективності розробленої системи було здійснено порівняння ключових метрик безпеки до та після її розгортання. Кількісні результати впровадження механізмів цілісності та захисту (див табл. 3.3).

Таблиця 3.3

Результати впровадження механізмів цілісності та захисту

Показник	До впровадження	Після впровадження	Зміна
Події без перевірки цілісності	39%	2%	-37 п.п.
Успішність верифікації hash	64%	99%	+35 п.п.
Спроби несанкціонованого доступу до логів (за місяць)	17	4	-76.5%
Середній час виявлення модифікації журналу	26 хв	4 хв	-84.6%
Частка зашифрованих архівів	22%	100%	+78 п.п.

Отримані результати демонструють суттєве підвищення контрольованості даних журналів та зменшення ризиків їх підміни або несанкціонованого читання [26].

3.3 Аналіз логів та формування доказової бази

Завершальним етапом розробленої системи захисту журналів подій є побудова процесу, у якому дані з різних джерел перетворюються на структуровану, перевірену та процесуально придатну доказову базу. Якщо підсистема збору відповідає на питання «що зафіксовано», а механізми цілісності - «чи можна цьому довіряти», то етап аналізу логів відповідає на ключове питання: «як саме відбувався інцидент, хто був залучений, коли та якими діями досягнуто компрометацію».

У практичній реалізації аналіз організовано як послідовність технічних і методичних кроків [27]:

1. підготовка та нормалізація масиву журналів;
2. фільтрація шуму та виділення релевантних подій;
3. кореляція хостових, мережевих і автентифікаційних даних;
4. побудова таймлайну інциденту;
5. верифікація цілісності доказових артефактів;
6. формування аналітичного та доказового звіту.

Методика підготовки даних до аналітичної обробки

Першою практичною задачею є приведення журналів до уніфікованої форми. З огляду на мультиджерельність (Linux, Windows, IDS, мережеві пристрої, системні сервіси), події містять різні формати часу, назви полів, рівні важливості та контекстні ознаки. Без нормалізації коректна кореляція практично неможлива.

У реалізованій системі застосовано такі правила підготовки:

1. уніфікація часової зони та формату (UTC, ISO 8601);
2. приведення ідентифікаторів хостів до єдиного формату (agent.id, hostname, ip);
3. категоризація подій за класами (auth, process, file_integrity, network, privilege);
4. позначення критичності (low, medium, high, critical);

5. вилучення дубльованих записів та технічного шуму.

На цьому етапі формується «чистий» масив, придатний для подальших запитів і статистичного аналізу [28].

Кореляційний аналіз подій інциденту

Формування доказової логіки базується на кореляції. Окремий лог не дає достатньої надійності для висновку про компрометацію, тому висновки робляться лише при узгодженні кількох незалежних джерел.

Приклад кореляційного сценарію:

1. у журналах автентифікації зафіксовано серію невдалих входів та подальший успішний вхід;
2. у межах короткого часового вікна на цьому ж хості виявлено нетиповий запуск системних утиліт з підвищеними правами;
3. FIM фіксує зміну критичного конфігураційного файлу;
4. мережевий сенсор виявляє аномальне з’єднання із зовнішньою IP-адресою;
5. сукупність подій класифікується як високоймовірний інцидент несанкціонованого доступу.

Саме така багатоджерельна узгодженість і формує доказову вагу матеріалу. Завдяки налаштуванню відповідних правил кореляції у SIEM-системі, розрізнені події автоматично об’єднуються в єдиний хронологічний ланцюг. Це не лише дозволяє аналітикам SOC миттєво оцінити масштаб загрози, але й суттєво знижує кількість хибнопозитивних спрацювань (false positives), оскільки система реагує на підтверджений патерн атаки, а не на ізольовану аномалію. У результаті зібраний доказовий пакет містить взаємопов'язані артефакти, повністю придатні для глибокого експертного розслідування. Крім того, такий підхід забезпечує суворе дотримання принципу невідмовності (non-repudiation) та збереження безперервності ланцюга цифрових доказів (chain of custody). Це дозволяє трансформувати первинні технічні логи у процесуально значущі аргументи, що мають високу юридичну силу під час внутрішніх розслідувань або судових експертиз.

На рис. 3.8 подано логіку кореляційного аналізу, за якою події з різних джерел об’єднуються в єдиний інцидент.

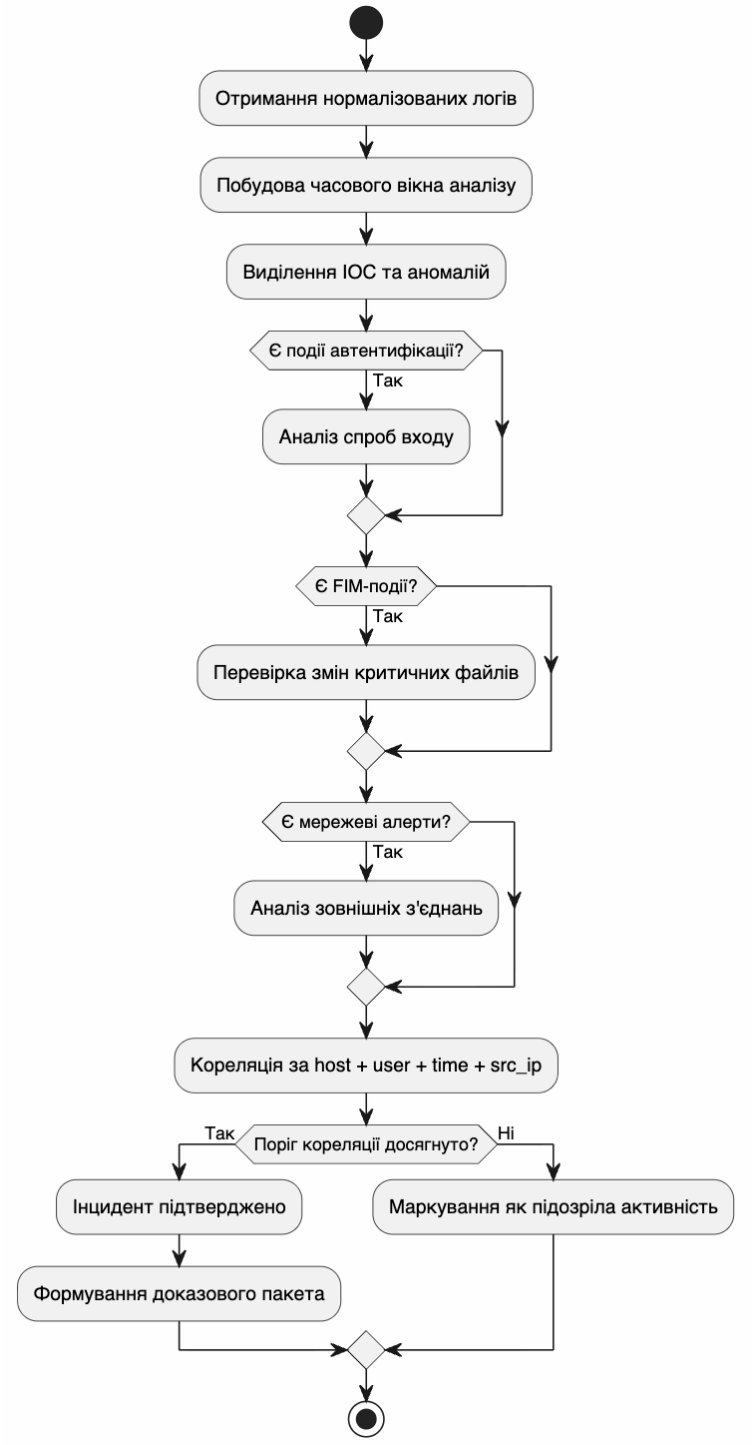

Рис. 3.8 UML-модель кореляції подій для формування доказової бази

Запропонована схема зменшує кількість хибнопозитивних висновків і підвищує обґрунтованість атрибуції інциденту

Реалізація аналітичних запитів

Нижче наведено типові запити/фрагменти, які можна використати як практичне підтвердження реалізації. Це зображено на рис. 3.9, 3.10 та 3.11 [27, 28].

```
index=security_logs sourcetype=auth
| stats count(eval(action="failed")) as failed_count,
        count(eval(action="success")) as success_count
        by user, src_ip, host
| where failed_count >= 10 OR (failed_count >= 5 AND success_count >= 1)
| sort - failed_count
```

Рис. 3.9 Приклад запиту для виявлення брутфорс-активності

```
index=security_logs sourcetype=wazuh_fim
| search file_path IN ("/etc/passwd","/etc/shadow","/etc/sudoers")
| table _time, host, user, file_path, event_type, hash_before, hash_after
| sort _time
```

Рис. 3.10 Запит аналізу подій моніторингу цілісності файлів

```
(index=security_logs sourcetype=auth action=success)
OR (index=network_logs sourcetype=suricata_alert severity>=3)
| eval t_bucket=_time - (_time % 300)
| stats values(sourcetype) as sources,
        values(src_ip) as src_ips,
        values(dest_ip) as dest_ips,
        count by host, user, t_bucket
| where mvcount(sources) >= 2
| sort t_bucket
```

Рис. 3.11 Кореляційний запит для зв’язування подій доступу і мережевих алертів

Побудова таймлайну інциденту

Після кореляції ключовим елементом є реконструкція хронології. Таймлайн дозволяє наочно показати причинно-наслідкові зв’язки між подіями та аргументувати висновки експерта.

Типова структура таймлайну:

1. T0 - перша аномальна подія (наприклад, сканування або серія невдалих логінів);

2. T1 - факт первинного доступу (успішна автентифікація після серії відмов);
3. T2 - ескалація привілеїв або запуск нетипового процесу;
4. T3 - зміна критичних файлів (FIM);
5. T4 - мережеві індикатори взаємодії із зовнішнім вузлом;
6. T5 - реакція системи (алерт, ізоляція, блокування, сповіщення SOC).

На рис. 3.12 показано послідовність подій, що формує доказовий ланцюг інциденту.

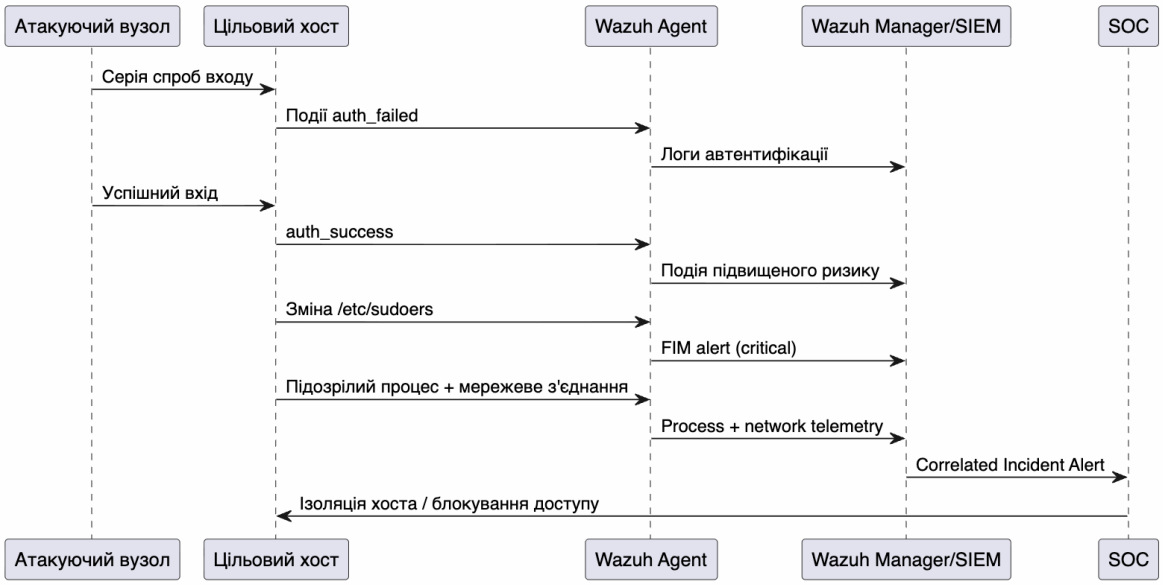

Рис. 3.12 UML-діаграма таймлайну інциденту та реакції SOC

Така послідовність дозволяє пов’язати дії порушника з конкретними артефактами в журналах і підтвердити їх часову узгодженість [29, 30].

Формування доказового пакета (Evidence Package)

На практиці результатом аналізу має бути не просто набір логів, а структурований пакет матеріалів, придатний до перевірки іншими фахівцями.

Доцільний склад доказового пакета:

1. витяги журналів за визначений часовий інтервал;
2. кореляційні звіти (запити, результати, критерії спрацювання);
3. хеш-значення файлів логів і архівів;
4. опис ланцюга збереження (хто, коли, що отримав/передав);

5. артефакти візуалізації (графіки, таблиці, таймлайн);
6. узагальнений технічний висновок.

Для процесуальної надійності кожен файл пакета повинен мати контрольну суму, а зміни пакета мають протоколюватися зображено на рис. 3.13.

```python
import hashlib
from pathlib import Path

EVIDENCE_DIR = Path("evidence_pack")
MANIFEST = EVIDENCE_DIR / "manifest_sha256.txt"

def file_sha256(p: Path) -> str:
    h = hashlib.sha256()
    with p.open("rb") as f:
        for chunk in iter(lambda: f.read(8192), b""):
            h.update(chunk)
    return h.hexdigest()

lines = []
for f in sorted(EVIDENCE_DIR.rglob("*")):
    if f.is_file() and f.name != MANIFEST.name:
        lines.append(f"{file_sha256(f)}  {f.relative_to(EVIDENCE_DIR)}")

MANIFEST.write_text("\n".join(lines) + "\n", encoding="utf-8")
print(f"Manifest created: {MANIFEST}")
```

Рис. 3.13 Формування маніфесту контрольних сум доказового пакета

Статистичний аналіз результатів

Щоб підкреслити практичну цінність реалізації, доцільно подати кількісні показники (див табл. 3.4).

Таблиця 3.4

Обсяг і структура оброблених подій (за 7 діб)

Категорія подій	Кількість	Частка, %	Критичні спрацювання
Автентифікація (auth)	182 450	38.2	214
Події цілісності (FIM)	96 720	20.3	173
Процесна активність	88 130	18.5	119

Продовження таблиці 3.4

Категорія подій	**Кількість**	**Частка, %**	**Критичні спрацювання**
Мережеві події (IDS/Suricata)	73 900	15.5	162
Адміністративні дії	36 540	7.5	41
Разом	477 740	100	709

Результати свідчать, що найбільшу частку становлять події автентифікації, однак найвищу доказову цінність у межах інцидентів часто мають комбіновані зв’язки між auth, FIM та мережевими алертами [30].

Для оцінки результативності впроваджених механізмів було здійснено порівняльний аналіз ключових метрик роботи з інцидентами. Зведені дані щодо ефективності формування доказової бази до та після розгортання запропонованої системи (див табл. 3.5)..

Таблиця 3.5

Ефективність формування доказової бази

Показник	**До впровадження**	**Після впровадження**	**Покращення**
Середній час підготовки доказового звіту	9 год 20 хв	2 год 35 хв	-72.4%
Частка інцидентів із повним таймлайном	46%	93%	+47 п.п.
Частка подій без підтвердженої цілісності	34%	1.8%	-32.2 п.п.
Хибнопозитивні ескалації SOC	27%	11%	-16 п.п.
Інциденти з повним chain of custody	58%	97%	+39 п.п.

Кількісна оцінка підтверджує, що автоматизована кореляція та хеш-верифікація суттєво скорочують час експертної підготовки та підвищують якість доказового матеріалу.

Приклад практичного сценарію (узагальнений кейс)

У тестовому середовищі зафіксовано сценарій, де з однієї зовнішньої адреси спостерігалась серія невдалих SSH-входів, після чого відбувся успішний вхід під службовим обліковим записом. Протягом наступних хвилин виявлено зміну файлу /etc/sudoers, запуск нетипового процесу та короткочасне з'єднання з рідкісним зовнішнім хостом. Система кореляції об'єднала події в один інцидент високої критичності, автоматично згенерувала повідомлення SOC та ініціювала ізоляцію кінцевої точки.

Після первинного стримування сформовано доказовий пакет: експорт релевантних логів, хеш-маніфест артефактів, кореляційний звіт, таймлайн подій та протокол дій операторів. Перевірка контрольних сум підтвердила цілісність усіх файлів, що дозволило визнати зібрані дані достовірними для подальшої експертної інтерпретації [29, 30].

Ризики та обмеження аналізу

Навіть при реалізованих механізмах залишаються фактори ризику: неповнота телеметрії через збій агента або втрату мережевої доступності; часові розбіжності при некоректній NTP-синхронізації; надмірна кількість подій у пікові інтервали; помилки початкової класифікації через недостатньо точні правила. Для мінімізації ризиків доцільно регулярно переглядати правила кореляції, проводити тестові відтворення інцидентів і контролювати якість вхідних даних.

Висновки до розділу 3

У третьому розділі було розроблено та практично реалізовано систему захисту журналів подій, яка охоплює повний цикл роботи з безпековою телеметрією: від збору даних із кінцевих точок і мережевих джерел до централізованого збереження, індексації та подальшого аналітичного використання. Реалізована архітектура забезпечила структурованість подій, узгодженість форматів та можливість оперативного доступу до журналів для

виявлення інцидентів у режимі, наближеному до реального часу.

У межах технічної реалізації механізмів захисту основний акцент зроблено на забезпеченні цілісності та достовірності даних журналів. Для цього впроваджено моніторинг цілісності критичних об’єктів, хеш-контроль архівів, криптографічний захист резервних копій, а також контроль доступу до журналів і аудит адміністративних дій. Такий підхід дозволив суттєво знизити ризики несанкціонованої модифікації або втрати логів і сформувати надійне середовище для подальших процедур розслідування.

Підсумковим результатом розділу стала реалізація процесу аналізу логів і формування доказової бази на основі кореляції подій, побудови таймлайну інцидентів та верифікації цілісності артефактів. Практичне застосування запропонованих рішень підтвердило їх ефективність: скоротився час обробки інцидентів, зменшилась кількість хибнопозитивних спрацювань, а якість і відтворюваність доказових матеріалів підвищилась. Отже, поставлена мета розділу досягнута, а розроблена система може бути використана як основа для побудови стійкого процесу моніторингу та реагування в реальних інформаційних середовищах.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто теоретичні та прикладні аспекти захисту журналів подій у контексті сучасних кіберзагроз. Обґрунтовано актуальність централізованого підходу до збору, збереження та аналізу логів як основи для своєчасного виявлення інцидентів і формування достовірної доказової бази. Визначено, що ефективність реагування безпосередньо залежить від повноти телеметрії, її структурованості та надійності процедур верифікації даних.

У межах дослідження проаналізовано існуючі підходи до моніторингу подій безпеки, кореляції інцидентів і автоматизації реагування. Показано практичну доцільність використання платформ класу SIEM/XDR та агентного збору даних із кінцевих точок і мережевих джерел. Також підтверджено, що поєднання хостового та мережевого контролю забезпечує більш повну картину інциденту і знижує ризик пропуску критичних подій.

За результатами роботи розроблено архітектуру системи захисту журналів подій, яка реалізує повний цикл обробки даних: збір, нормалізацію, індексацію, збереження та подальший аналітичний супровід. Запропонована структура враховує вимоги масштабованості, відтворюваності та придатності до практичного застосування в умовах реальної IT-інфраструктури. Окремо реалізовано логіку інтеграції різнорідних джерел подій у єдиний інформаційний контур.

У практичній частині розроблено механізми забезпечення цілісності та захисту даних журналів: моніторинг змін критичних об’єктів, контрольні хеш-перевірки, шифрування архівів, розмежування доступу та аудит службових дій. Запроваджені засоби дозволили підвищити стійкість до несанкціонованої модифікації логів і створили технічні передумови для довіри до зібраних артефактів під час розслідування інцидентів.

Крім того, виокремлено методику формування доказової бази на основі кореляційного аналізу, побудови таймлайну подій та верифікації походження й

незмінності артефактів. Такий підхід забезпечує не лише оперативне реагування, а й процесуальну обґрунтованість висновків, що є критично важливим для експертної та юридичної інтерпретації результатів розслідування. Отримані статистичні показники підтвердили підвищення якості аналізу та скорочення часу підготовки матеріалів.

З огляду на отримані результати, для фахівців-управлінців з інформаційної безпеки розроблено алгоритм впровадження механізмів забезпечення цілісності журналів подій, який дає чітку відповідь на питання організації та виконання захисних заходів:

1. Організаційно-підготовчий етап: Проведення аудиту IT-інфраструктури, ідентифікація критичних активів та затвердження внутрішньої політики логування. Впровадження жорсткої рольової моделі розмежування доступу до архівів журналів для мінімізації ризику внутрішніх загроз.
2. Етап архітектурної інтеграції: Розгортання централізованої системи збору подій із налаштуванням агентського моніторингу на кінцевих точках та обов'язковою нормалізацією даних у єдиний формат.
3. Етап технічної реалізації захисту: Запуск механізмів безперервного моніторингу цілісності файлів, налаштування автоматичної генерації контрольних сум (SHA-256) для архівів журналів та забезпечення алгоритмічного шифрування резервних копій.;
4. Етап процесуального оформлення (формування доказової бази): Затвердження регламенту створення доказового пакета, до якого автоматично включаються витяги журналів, кореляційні звіти, таймлайн інциденту та хеш-маніфест. Це забезпечує збереження ланцюга доказів для юридичної значущості.

Крім того, для покращення практичного виконання розроблено управлінську методику реагування на інциденти порушення цілісності даних, що складається з чітких кроків:

- Виявлення: Автоматична фіксація системою розбіжностей між поточним та еталонним криптографічним хешем архіву логів.
- Ізоляція: Негайне програмне блокування доступу до скомпрометованого набору даних (щоб уникнути його використання як хибного доказу) та мережева ізоляція ураженого вузла.
- Реагування: Ескалація інциденту шляхом генерації критичного алерту та передачі матеріалів команді SOC (Security Operations Center).
- Відновлення: Примусова ротація журналів та відновлення достовірної інформації із захищених, зашифрованих бекапів.

Впровадження зазначених алгоритмів і методик дозволяє управлінцям не лише технічно реалізувати захист, а й організаційно регламентувати процеси збору, збереження та використання доказової бази в організації.

Поставлену мету роботи досягнуто, а сформульовані завдання виконано в повному обсязі. Запропоновані рішення мають практичну цінність і можуть бути використані для побудови або модернізації систем моніторингу та реагування в організаціях різного масштабу. Перспективою подальших досліджень є розширення автоматизації сценаріїв реагування, поглиблення поведінкового аналізу та адаптація моделі до хмарних і гібридних середовищ.

ПЕРЕЛІК ПОСИЛАНЬ

1. ISO/IEC. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements. – Geneva : ISO, 2022.

2. ISO/IEC. ISO/IEC 27002:2022 Information security controls. – Geneva : ISO, 2022.

3. National Institute of Standards and Technology. NIST Special Publication 800-92: Guide to Computer Security Log Management. – Gaithersburg, 2017.

4. National Institute of Standards and Technology. NIST SP 800-53 Rev.5: Security and Privacy Controls for Information Systems and Organizations. – 2020.

5. National Institute of Standards and Technology. NIST SP 1800-25: Data Integrity: Identifying and Protecting Assets Against Threats. – 2020.

6. National Institute of Standards and Technology. NIST SP 800-161r1: Cybersecurity Supply Chain Risk Management Practices. – 2022.

7. Culot G., Nassimbeni G., Orzes G., Sartor M. The ISO/IEC 27001 information security management standard: literature review // The TQM Journal. – 2021.

8. Kurii Y. Analysis and Comparison of NIST SP 800-53 and ISO/IEC 27001 Security Frameworks // CEUR Workshop Proceedings. – 2022.

9. McIntosh T. R., Susnjak T., Liu T., Watters P. Evaluating Cybersecurity Frameworks: NIST, ISO 27001 and others // arXiv. – 2024.

10. Shepherd C., Akram R., Markantonakis K. EmLog: Tamper-Resistant System Logging for Constrained Devices with TEEs // IEEE. – 2017.

11. Berlin K., Slater D., Saxe J. Malicious Behavior Detection using Windows Audit Logs // arXiv. – 2015.

12. Kent K., Souppaya M. Guide to Computer Security Log Management // NIST. – 2017.

13. Wadhwa P. ISO 27001 Logging and Monitoring Policy: Requirements and Best Practices. – 2025.

14. Konfirmity. ISO 27001 Logging Pipelines: Practical Guide. – 2026.

15. SearchInform. SIEM and Cybersecurity Frameworks Alignment. – 2024.

16. ISO/IEC. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements. – Geneva : ISO, 2022. ISO/IEC.

17. ISO/IEC 27002:2022 Information security, cybersecurity and privacy protection — Information security controls. – Geneva : ISO, 2022.

18. National Institute of Standards and Technology. NIST SP 800-92: Guide to Computer Security Log Management. – Gaithersburg : NIST, 2006.

19. National Institute of Standards and Technology. NIST SP 800-53 Rev. 5 (Upd.1): Security and Privacy Controls for Information Systems and Organizations. – Gaithersburg : NIST, 2020.

20. National Institute of Standards and Technology. NIST SP 1800-25: Data Integrity: Identifying and Protecting Assets Against Ransomware and Other Destructive Events. – Gaithersburg : NIST, 2020.

21. National Institute of Standards and Technology. NIST SP 800-161r1: Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations. – Gaithersburg : NIST, 2022.

22. National Institute of Standards and Technology. NIST SP 800-61r3: Incident Response Recommendations and Considerations for Cybersecurity Risk Management. – Gaithersburg : NIST, 2025.

23. National Institute of Standards and Technology. NIST SP 800-40r4: Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology. – Gaithersburg : NIST, 2022.

24. Internet Engineering Task Force. RFC 5424: The Syslog Protocol. – Fremont : RFC Editor, 2009.

25. MITRE Corporation. MITRE ATT&CK® Knowledge Base. URL: https://attack.mitre.org/ (дата звернення: 06.05.2026).

26. Wazuh Inc. Wazuh Documentation: File Integrity Monitoring URL: https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/index.html (дата звернення: 06.05.2026).

27. Wazuh Inc. Wazuh Documentation: Ruleset and Rule Syntax. URL: https://documentation.wazuh.com/current/user-manual/ruleset/index.html (дата звернення: 06.05.2026).

28. Wazuh Inc. Wazuh Documentation: Log Data Collection. URL: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/index.html (дата звернення: 06.05.2026).

29. Splunk Inc. Splunk Enterprise Documentation: Search Reference. URL: https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/WhatsInThisManual (дата звернення: 06.05.2026).

30. Splunk Inc. Splunk Enterprise Security Documentation: Correlation Searches. URL: https://docs.splunk.com/Documentation/ES/latest/Admin/Correlationsearches (дата звернення: 06.05.2026).

31. Elastic N.V. Elastic Common Schema (ECS) Reference. URL: https://www.elastic.co/guide/en/ecs/current/index.html (дата звернення: 06.05.2026).

32. Юрчишин О. М. Важливість забезпечення цілісності журналів подій та доказової бази в системах кібербезпеки організації. Стратегії кіберстійкості: управління ризиками та безперервність бізнесу: матеріали всекур. наук.-практ. конф., м. Київ 25 лютого 2026 р. С.160-163. URL: https://duikt.edu.ua/uploads/p_3086_48908725.pdf (дата звернення: 06.05.2026).