МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Кваліфікаційна наукова
праця на правах рукопису

ЖИГАРЕВИЧ ОКСАНА КОСТЯНТИНІВНА

УДК 004.056:004.057.3 (043.3)

ДИСЕРТАЦІЯ
СИСТЕМА КОРЕЛЮВАННЯ ПОДІЙ ТА УПРАВЛІННЯ ІТ-ІНЦИДЕНТАМИ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

Спеціальність 05.13.06 – «Інформаційні технології»

Подається на здобуття наукового ступеня кандидата технічних наук

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

О.К. Жигаревич

Науковий керівник:
Сидоренко Вікторія Миколаївна
кандидат технічних наук, доцент

Київ – 2026

АНОТАЦІЯ

Жигаревич О.К. Система корелювання подій та управління IT-інцидентами на об’єктах критичної інфраструктури. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 «Інформаційні технології». – Державний університет «Київський авіаційний інститут», Київ, 2026.

Проведено аналіз сучасних підходів до управління IT-інцидентами на об’єктах критичної інфраструктури (ОКІ) для виявлення їх переваг та недоліків. За результатами проведеного аналізу підходів до виявлення аномалій в хмарному середовищі встановлено, що кожен метод виявлення має свої переваги та працює краще для певних наборів даних, але жоден не є універсальним і не може виявити всі сто відсотків шкідливих програм. Аналіз сучасних типів баз даних, що використовуються в SIEM-системах, показав, що кожен з їх видів залишається актуальним у власній сфері, де взаємозв'язки між даними обумовлені конкретною структурою СУБД, а використання однієї бази даних для всіх цих задач не відповідає вимогам архітектури та безпеки. Крім того, проаналізовані існуючі на ринку рішення інтеграційних шин даних, та встановлено, що кожні з них мають свої особливості та суттєві відмінності, які визначають їх сферу використання, а також відрізняються функціоналом, додатковими налаштуваннями та вартістю ліцензії. Крім того, систематизовано та представлено детальний аналіз 16 SIEM-систем за 18 запропонованими критеріями. Зокрема відображено їх функціональність, основний принцип роботи, а також проведено порівняльний аналіз їх можливостей та відмінностей, переваг та недоліків використання, та відповідності до міжнародних специфікацій та стандартів. Проведений аналіз дозволив формалізувати завдання

дисертаційного дослідження щодо розроблення моделей та системи корелювання подій та управління ІТ-інцидентами на ОКІ.

Удосконалено структурно-аналітичну модель оброблення даних, яка завдяки формулюванню команд для передачі керування програмному клієнту ІКТ, додатковій обробці метаданих у хмарній системі та інтелектуалізованому виявленню сигнатур, дозволяє підвищити ефективність виявлення аномалій в хмарних ІКС. Експериментальне дослідженя моделі дозволило оцінити часові характеристики обробки одного елемента метаданих та розробити керуючі команди. Її відмінною особливістю є врахування необхідності формування команд передачі управління програмному клієнту ІКС, що загалом підвищило точність результатів оцінки часових характеристик до 1,7 разів, і характеристик спотворень (затримок) до 4,5 разів. Крім того, на основі дата сету NSL-KDD навчено нейронну мережу з точки зору виявлення аномалій типу DoS, U2R, R2L та Probe.

Розроблено модель онтологіко-реляційного сховища даних, яка за рахунок попередньої індексації та синтезу двох різних баз даних з відповідними характеристиками, дає можливість покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість пошуку та отримання великих обсягів інформації. За результатами експериментальних досліджень обґрунтовано вибір найбільш ефективних систем керування базами даних – Elasticsearch та MongoDB, які відповідають визначеній множині критеріїв і забезпечують розв’язання поставлених задач. Крім того, розроблено методику, яка дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірності та забезпечувати високу швидкість пошуку.

Удосконалено модель інтеграційної шини даних (ІШД), яка за рахунок декомпозиції функціональності сервісів (кожен з яких відповідає за окреме завдання і може працювати ізольовано від інших) та визначення критичності сервісів, дозволяє розподілити навантаження на послуги та

гарантує безперервність обміну даними для ефективного функціонування системи корелювання подій та управління ІТ-інцидентами. Встановлено, що ІШД виступає централізованим компонентом, який забезпечує інтеграцію сервісів, маршрутизацію та перетворення даних, а також їх узгоджену взаємодію через сервісні інтерфейси. У межах експериментального дослідження визначено значення рангів критичності сервісів для найбільш ефективних баз даних – MongoDB та Elasticsearch. Отримані результати дозволили сформувати пріоритетну чергу обслуговування сервісів і забезпечити оптимальний розподіл навантаження в ІШД. На основі розробленої моделі, було сформовано відповідну специфікацію реалізації SIEM-систем на ОКІ, у вигляді основних та додаткових вимог.

Отримала подальший розвиток система корелювання подій та управління ІТ-інцидентами на ОКІ, яка за рахунок використання розроблених моделей обробки даних, онтологіко-реляційного сховища даних та інтеграційної шини даних, дає змогу формалізувати інформаційну технологію, що реалізує процеси управління ІТ-інцидентами на ОКІ згідно вимог міжнародних стандартів та найкращих світових практик щодо створення систем управління ІТ-інцидентами. Експериментальне дослідження розробленої інформаційної технології підтвердило її відповідність сучасним вимогам до систем управління ІТ-інцидентами. Зокрема, забезпечено централізоване управління компонентами системи, візуалізацію даних через відповідні інтерфейси, підтримку відкритого програмного інтерфейсу (API), механізми аутентифікації та авторизації, а також реалізовано властивості масштабованості, відмовостійкості, збору та фільтрації подій і управління обліковими записами. Розроблена система може використовуватися для ефективного управління ІТ-інцидентами на ОКІ, що впливають на КВР. Крім того, створено спеціалізований програмний застосунок, призначений для реалізації зазначених функцій управління інцидентами.

На основі розробленої інформаційної технології та спеціалізованого програмного забезпечення проведено експериментальне дослідження, у результаті якого верифіковано розроблені в роботі моделі та систему. Результати дисертаційного дослідження впроваджені і використовуються у діяльності ТОВ «AxxonSoft» (акт про впровадження від 11.03.2026), НДЛ протидії кіберзагрозам авіаційної галузі КАІ (акт про впровадження від 12.02.2024), а також у навчальному процесі кафедри комп'ютерних наук та кібербезпеки Волинського національного університету імені Лесі Українки для підвищення ефективності підготовки фахівців з ІТ (акт про впровадження від 21.12.2023).

Ключові слова: ІТ-інцидент, критична інфраструктура, об’єкти критичної інфраструктури, інформаційний об'єкт, виявлення аномалій, виявлення вразливостей, види аномалій, хмарні системи, онтологія, підтримка рішень, SIEM, система корелювання подій та управління ІТ-інцидентами.

ABSTRACT

Zhyharevych O. System for events correlation and IT-incident management in critical infrastructure objects. – Manuscript.

Thesis for a Candidate of Technical Science degree in specialty 05.13.06 «Information technology». – State University «Kyiv Aviation Institute», Kyiv, 2026.

An analysis of modern approaches to IT incident management in critical infrastructure (CI) was carried out in order to identify their advantages and disadvantages . The analysis of anomaly detection methods in cloud environments showed that each method has its own advantages and performs better for specific datasets; however, none of them is universal and capable of detecting all types of malicious software. The analysis of modern database types used in SIEM systems demonstrated that each type remains relevant within its own domain, where relationships between data are determined by the specific structure of the database management system. At the same time, the use of a single database for all tasks does not meet architectural and security requirements. In addition, existing data integration bus solutions were analyzed, revealing their specific features and significant differences that determine their application domains, functionality, configuration capabilities, and cost. Furthermore, a comprehensive analysis of 16 SIEM systems was performed based on 18 criteria, including their functionality, operating principles, advantages, disadvantages, and compliance with international standards. The conducted analysis allowed formalizing the objectives of the dissertation research aimed at developing models and a system for event correlation and IT incident management in CI.

An improved structural-analytical data processing model is proposed, which, due to the formulation of control commands for interaction with the client application, additional metadata processing in the cloud environment, and intelligent signature detection, increases the efficiency of anomaly detection in cloud information and communication systems. Experimental studies made it

possible to evaluate the time characteristics of processing a single metadata element and to develop control commands. A distinctive feature of the model is the consideration of control command generation for the client application, which improved the accuracy of time characteristic estimation by up to 1.7 times and delay characteristics by up to 4.5 times. In addition, a neural network was trained using the NSL-KDD dataset to detect DoS, U2R, R2L, and Probe attack types.

A model of an ontology-relational data warehouse has been developed, which, through preliminary indexing and the combination of two different types of databases with appropriate characteristics, improves data storage and classification efficiency and ensures high-speed search and processing of large data volumes. Based on experimental results, Elasticsearch and MongoDB were justified as the most efficient database management systems that satisfy the defined criteria and support the required tasks. In addition, a methodology was developed that enables the indexing service to interact with external data repositories, perform scaling, aggregation, and analysis, identify patterns, and provide high-performance search capabilities.

The model of the enterprise service bus (ESB) has been improved. Due to the decomposition of service functionality (each service performs a specific task and operates independently) and the determination of service criticality, the model ensures effective load distribution and uninterrupted data exchange for efficient operation of the event correlation and IT incident management system. It has been established that the ESB acts as a centralized component that provides service integration, data routing and transformation, as well as coordinated interaction through service interfaces. Within the experimental study, the criticality ranks of services for the most efficient databases (MongoDB and Elasticsearch) were determined. The obtained results made it possible to define service prioritization and ensure optimal load balancing in the ESB. Based on the developed model, a specification for the implementation of SIEM systems in CI was formed in the form of basic and additional requirements.

The event correlation and IT incident management system in critical infrastructure has been further developed. Through the use of the proposed data processing models, the ontology-relational data warehouse, and the enterprise service bus, it enables the formalization of an information technology that implements IT incident management processes in critical infrastructure in accordance with international standards and best global practices for building incident management systems. Experimental evaluation of the developed information technology confirmed its compliance with modern requirements for IT incident management systems. In particular, centralized management of system components and functionalities has been ensured, along with data visualization through appropriate interfaces, support for an open application programming interface (API), implementation of authentication and authorization mechanisms, as well as scalability, fault tolerance, event collection and filtering, and account management capabilities. The developed system can be used for effective management of IT incidents occurring in critical infrastructure and affecting critical assets. In addition, a specialized software application has been developed to support the implementation of the described incident management functions.

Based on the developed information technology and specialized software, experimental research was conducted, which confirmed the correctness and efficiency of the developed models and system. The results of the dissertation have been implemented and are used in the activities of AxxonSoft LLC, the Research Laboratory for Cyber Threat Counteraction in the Aviation Industry (KAI), and in the educational process of the Department of Computer Science and Cybersecurity of Lesya Ukrainka Volyn National University.

Keywords: IT incident, critical infrastructure, critical infrastructure objects, information object, anomaly detection, vulnerability detection, types of anomalies, cloud systems, ontology, decision support, SIEM, event correlation and IT incident management system.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях України:

1. Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., Смірнова Т.В. Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури. *Кібербезпека: освіта, наука, техніка*. 2023. Т. 3. № 19. С. 176-196. DOI: https://doi.org/10.28925/2663-4023.2023.19.176196

2. Гнатюк С.О., Бердибаєв Р.Ш., Богун А.М., Сидоренко В.М., Положенцев А.А., Жигаревич О.К. Інтеграційна шина даних для ефективного функціонування системи управління подіями інформаційної безпеки. *Проблеми інформатизації та управління*. 2023. Т. 3. № 75. С. 29-40. DOI: https://doi.org/10.18372/2073-4751.75.18014

3. Жигаревич О.К., Бердибаєв Р.Ш., Сидоренко В.М., Положенцев А.А., Кримська А.О. Модель онтологіко-реляційного сховища даних для функціонування хмарної SIEM-системи. *Проблеми інформатизації та управління*. 2023. Т. 4. № 76. С. 17-27. DOI: https://doi.org/10.18372/2073-4751.76.18236.

4. Дмитрук Я.В., Гришанович Т.О., Глинчук Л.Я., Жигаревич О.К. Кібервійна як різновид інформаційних війн. Захист кіберпростору України. *Кібербезпека: освіта, наука, техніка*. 2022. Т. 4. №16. С. 28-36. DOI: https://doi.org/10.28925/2663-4023.2022.16.2836.

5. Жигаревич О.К., Медведєв М.В. Інформаційна система «Студент-ФКНІТ» засобами PHP. *Комп'терно-інтегровані технології: освіта, наука, виробництво*. 2017. № 26. С. 88-92.

6. Жигаревич О.К., Котлярець В.В., Луць А.Р. Модель екосистеми навчального програмного забезпечення. *Комп'терно-інтегровані технології: освіта, наука, виробництво*. 2017. № 26. С. 167-177.

7. Жигаревич О.К., Мельник В.М., Мельник К.В. Підтримка

оголошеної/встановленої комунікації в мережі через стандартні сокети API. *Комп'терно-інтегровані технології: освіта, наука, виробництво*. 2015. № 19. С. 23-27.

8. Мельник К.В., Мельник В.М., Багнюк Н.В., Жигаревич О.К., Климюк М. Система попереднього відбору кандидатів на основі нечіткої логіки. *Комп'терно-інтегровані технології: освіта, наука, виробництво*. 2015. № 19. С. 114-120.

Статті в іноземних виданнях:

9. Pobochenko L., Prokopieva A., Zhyharevych O., Gavrylko O., Panikar G., Gavrilko T. Risks of investing in FinTech at the global and national levels. *CEUR Workshop Proceedings. Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2025)*, June 20 - 22, 2025, Kyiv, Ukraine, 2025. Vol. 4024. P. 468-478. URL: https://ceur-ws.org/Vol-4024/paper30.pdf. (Scopus) Q4, ISSN 1613-0073.

10. Sydorenko V., Zhyharevych O., Berdybaev R., Polozhentsev A., Fesenko A. Ontological-Relational Data Store Model for a Cloud-based SIEM System Development. *CEUR Workshop Proceedings. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2024),* February 28, 2024, Kyiv, Ukraine, 2024. Vol. 3654. P. 343-354. URL: https://ceur-ws.org/Vol-3654/paper29.pdf (*Scopus*) Q4, ISSN 1613-0073.

11. Gnatyuk S., Berdibayev R., Aleksander M., Sydorenko V., Zhyharevych O., Polozhentsev A. Software System for Cybersecurity Events Correlation and Incident Management in Critical Infrastructure. *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*. 2024. Vol. 213. P. 247-269. Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-62213-7_12. (*Scopus*) Q3, ISSN 2367-4512.

12. Zdolbitska N., Ostapchuk O., Lavrenchuk S., Terletskyi T., Kaidyk O., Zhyharevych O. Business information system for forecasting raw material stocks for

the production of flexible packaging. *2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Athens, Greece, 2024, P. 1-8. doi: 10.1109/DESSERT65323.2024.11122240. (*Scopus*), Q4.

13. Polozhentsev A., Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O. Novel Cyber Incident Management System for 5G-based Critical Infrastructures. *IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany, 2023, P. 1037-1041. DOI: 10.1109/IDAACS58523.2023.10348645. (Scopus), Q4, ISSN 2770-4262.

14. Gnatyuk S., Satybaldiyeva F., Sydorenko V., Zhyharevych O., Polozhentsev A. Model of Information Technology for Efficient Data Processing in Cloud-based Malware Detection Systems of Critical Information Infrastructure. *CEUR Workshop Proceedings, Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2023)*, February 28, 2023, Kyiv, 2023, Vol. 3421, P. 206-213. URL: https://ceur-ws.org/Vol-3421/short6.pdf (Scopus) Q4, ISSN 1613-0073.

15. Gnatyuk S., Zhaksigulova D., Zhyharevych O., Ospanova D., Chuba I. Studies on WSN Models for IoT-based Monitoring Systems in the Critical Infrastructure of the State. *CEUR Workshop Proceedings, Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2023-II),* October 26, 2023, Kyiv, 2023, Vol. 3550, P. 167-180. URL: https://ceur-ws.org/Vol-3550/paper14.pdf (*Scopus*) Q4, ISSN 1613-0073.

16. Smirnov O., Sydorenko V., Aleksander M., Zhyharevych O., Yenchev S. Simulation of the cloud IoT-based monitoring system for critical infrastructures. *CEUR Workshop Proceedings, Proceedings of the 2nd International Conference on Conflict Management in Global Information Networks (CMiGiN 2022)*, November 30, 2022, Kyiv, 2023, Vol. 3530, P. 256-265. URL: https://ceur-ws.org/Vol-3530/paper25.pdf (*Scopus*) Q4, ISSN 1613-0073.

17. Gnatyuk S., Sydorenko V., Yudin O., Zhyharevych O., Polozhentsev

A. Method for Calculating the Criticality Level of Sectoral Information and Telecommunication Systems. *CEUR Workshop Proceedings*, Proceedings of the: Information Technology and Implementation (IT&I2022), November 30 - December 02, 2022, Kyiv, Ukraine, Vol. 3347, Paper 20, P.234-245. URL: https://ceur-ws.org/Vol-3347/Paper_20.pdf (*Scopus*) Q4, ISSN 1613-0073.

18. Melnyk V., Bahnyuk N., Melnyk K., Zhyharevych O., Panasyuk N. Implementation of the simplified communication mechanism in the cloud of high performance computations. *East-European journal of Enterprise Technologies*. Kharkiv, 2017. No 2/2/86. P. 24-32. DOI: 10.15587/1729-4061.2017.98896 (*Scopus*) Q3, ISSN 1729-3774.

19. Melnyk V., Pekh P., Melnyk K., Bahnyuk N., Zhyharevych O. Design and implementation of interdomain communication mechanism for high performance data processing, *East-European journal of Enterprise Technologies*. Kharkiv, 2016. No 1(9). P. 10-15. DOI: 10.15587/1729-4061.2016.60629 (Scopus) Q3, ISSN 1729-3774.

Публікації, які додатково відображають наукові результати дисертації:

20. Сидоренко В., Положенцев А., Юдін О., Жигаревич О. Функціональна модель визначення критичності галузевих інформаційно-телекомунікаційних систем. *ABIA-2023:* XVI міжнар. наук.-техніч. конф., 18-20 квітня 2023 р.: тези доп., Київ: НАУ, 2023. С. 16.14-16.17.

21. Здолбіцька Н.В., Ліщина Н.М., Лавренчук С.В., Давиденко Н.В., Жигаревич О.К. Інтелектуальна інформаційна система «робот-гід». Матеріали Міжнародної наукової молодіжної школи *«Системи та засоби штучного інтелекту»* 28.11.2021р. Київ, 2021. С. 19-21.

22. Жигаревич О.К., Сидоренко В.М., Положенцев А.А., Сидоренко С.Ю. Модель онтологіко-реляційного сховища даних для функціонування хмарної SIEM-системи». *Кіберзахист особи, суспільства і держави*: наук.-практ. конф., с. Велятино, 24-27 січня 2024 р.: тези доп., Київ: НАУ, 2024. С. 14-15.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних;

ІБ – інформаційна безпека;

ІКС – інформаційно-комунікаційні системи;

ІКТ – інформаційно-комунікаційні технології;

ІС – інформаційна система;

ІТ – інформаційні технології;

ІШД – інтеграційна шина даних;

КВР – критично важливі ресурси;

КІ – критична інфраструктура;

КІІ – критична інформаційна інфраструктура;

КС – комп’ютерна система;

ОКІ – об’єкти критичної інфраструктури;

ОКІІ – об’єкти критичної інформаційної інфраструктури;

ПЗ – програмне забезпечення;

СУБД – система управління базами даних;

ШПЗ – шкідливе програмне забезпечення;

SIEM – security information and event management.

ВСТУП

Актуальність. Сучасна інформаційна інфраструктура складається з великої кількості систем та компонентів, що потребують постійного моніторингу та контролю. В таких умовах провідними державами світу приділяється значна увага питанням забезпечення безпеки та сталого функціонування життєво важливих об’єктів, до яких належать великі гідротехнічні споруди, мережі електростанцій, шкідливі хімічні виробництва, транспортні вузли, аеродроми тощо. Виведення таких об'єктів інфраструктури з ладу може призвести до швидких негативних, а іноді і катастрофічних наслідків відповідної інформаційно-комунікаційної системи (ІКС). Для забезпечення необхідного рівня протидії загрозам в автоматизованих системах та ІКС, а також для зниження ризику виникнення аналогічних інцидентів у майбутньому в Україні функціонує Урядова команда реагування на комп`ютерні надзвичайні події (CERT або CSIRT). Особливої уваги потребує процес управління та прийняття рішень щодо виявлення та усунення можливих загроз об’єктам критичної інфраструктури (ОКІ). Серед існуючої множини інструментів для ефективного реагування на інциденти CSIRT на ОКІ є використання SIEM-систем, функціонування яких полягає в оперативному збиранні, збереженні та аналітичній обробці даних про події безпеки, що першочергово формуються та фіксуються в системних журналах різних апаратних і програмних елементів, а також формують інформаційні інфраструктури: сервери, робочі станції, маршрутизатори, мережеві екрани, системи управління базами даних, системи виявлення атак, антивірусні засоби тощо. Основною метою таких систем є підвищення рівня цифрової стійкості ІКС за рахунок забезпечення можливості в режимі, близькому до реального часу, обробляти інформацію про безпеку та здійснювати корелювання подій і управління інформаційно-технологічними (ІТ) інцидентами. У межах даного дослідження ІТ-інцидент розглядається як

подія або набір подій в ІКС, що відображають відхилення від її нормального функціонування, зумовлені порушенням роботи сервісів, ресурсів чи ІТ процесів, і вимагають прийняття управлінських рішень.

Питаннями корелювання подій та управління ІТ-інцидентами у т.ч. на ОКІ займаються такі вітчизняні та закордонні вчені: Богачук І., Бурячок В., Соколов В., Aslan O., Berdibayev R., Karlzen H., Lee J., Pernul G., Vielberth M та інші. Проведений аналіз дозволив систематизувати існуючі SIEM-системи за їх функціональністю, основними принципами роботи, відповідністю до вимог міжнародних специфікацій і стандартів та інших запропонованих критеріїв. Було виділено перелік систем, які відповідають значній кількості критеріїв, проте відрізняються вартістю. Також визначено, що сьогодні доцільно використовувати open source системи з погляду витрат та можливості доповнення функціоналу під потреби конкретного підприємства ОКІ. З точки зору безпеки, найбільш придатним варіантом є розробка власної системи, яка матиме широкий функціонал з ІБ, буде гнучкою та масштабованою, а також захищеною від можливих вразливостей та бекдорів. Не зважаючи на велику кількість інструментальних рішень, жодне з них не вирішує всі наявні проблеми щодо управління ІТ-інцидентами. Для успішної реалізації заходів захисту ОКІ необхідне вирішення низки завдань, основне з яких пов'язане зі створенням єдиної системи моніторингу загроз безпеці, головною метою якої буде зниження до мінімального рівня ризику впливу на ОКІ і мінімізація збитків, що можуть виникнути внаслідок реалізації загроз.

З огляду на зазначене, розроблення системи корелювання подій та управління ІТ-інцидентами на ОКІ є *актуальною науково-технічною задачею*, що має теоретичне і практичне значення.

Зв’язок роботи з науковими програмами, планами, темами. Дисертаційна робота безпосередньо пов’язана з пріоритетними тематичними напрямами наукових досліджень і науково-технічних розробок, визначеними постановою Кабінету Міністрів України від 30 квітня 2024 р. № 476, та

відповідає напряму розвитку інформаційно-комунікаційних технологій (ІКТ), зокрема, інтелектуальні інформаційно-аналітичні системи, інтегровані системи баз даних і знань, національні інформаційні ресурси. Тематика дослідження також узгоджується зі Стратегією цифрового розвитку інновацій до 2030 року, зокрема в частині розвитку технологій штучного інтелекту, інтелектуального аналізу й оброблення даних. Теоретичні і практичні положення дисертаційної роботи було використано під час виконання науково-дослідної роботи у Державному університеті «Київський авіаційний інститут», а саме НДР «Методи, моделі та програмні засоби управління інцидентами кібербезпеки в критичній інфраструктурі держави» (д.р. № 0125U000624, 2025-2026 рр.).

Мета і задачі дослідження. Метою дисертаційної роботи є забезпечення можливості управління ІТ-інцидентами на ОКІ на основі розроблення та удосконалення моделей і синтезу системи управління інцидентами.

Для досягнення поставленої мети необхідно розв’язати такі **основні задачі:**

1) провести аналіз сучасних підходів до управління ІТ-інцидентами на ОКІ для виявлення їх переваг та недоліків;

2) удосконалити структурно-аналітичну модель оброблення даних для інтелектуалізованого виявлення аномалій у хмарних системах ІКС;

3) розробити модель онтологіко-реляційного сховища даних для зберігання та оброблення великих масивів інформації;

4) удосконалити модель інтеграційної шини даних для розподілу навантаження та захищеного обміну даними;

5) на основі запропонованих моделей розробити систему корелювання подій та управління ІТ-інцидентами на ОКІ;

6) створити спеціалізоване програмне забезпечення та провести верифікацію розроблених у роботі моделей та системи.

Об’єктом дослідження є процеси управління ІТ-інцидентами на ОКІ.

Предметом дослідження є моделі, системи і засоби управління ІТ-інцидентами на ОКІ.

Методи дослідження. Проведені дослідження базуються на сучасних методах: математичної логіки, на основі якої розроблено модель онтологіко-реляційного сховища даних; теорії множин, для формалізації сукупності різноманітних баз даних та чинникових ознак, у вигляді основних критеріїв відбору; теорії штучного інтелекту, на основі якої, відбувалось навчання нейронної мережі для виявлення аномалій дата сету NSL-KDD; теорії комп’ютерних мереж, для розробки моделі інтеграційної шини даних; теорії системного та структурного аналізу, для представлення моделі оброблення даних. та обробки результатів експериментів і верифікації ефективності розроблених моделей та системи.

Наукова новизна одержаних результатів полягає у такому:

– *удосконалено* структурно-аналітичну модель оброблення даних, яка завдяки формулюванню команд для передачі керування програмному клієнту ІКТ, додатковій обробці метаданих у хмарній системі та інтелектуалізованому виявленню сигнатур, дозволяє підвищити ефективність виявлення аномалій в хмарних ІКС;

– *вперше розроблено* модель онтологіко-реляційного сховища даних, яка за рахунок попередньої індексації та синтезу двох різних баз даних (Elasticsearch та MongoDB) з відповідними характеристиками, дає можливість покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість пошуку та отримання великих обсягів інформації;

– *удосконалено* модель інтеграційної шини даних, яка за рахунок декомпозиції функціональності сервісів (кожен з яких відповідає за окреме завдання і може працювати ізольовано від інших) та визначення критичності сервісів, дозволяє розподілити навантаження на послуги та гарантує

безперервність обміну даними для ефективного функціонування системи корелювання подій та управління ІТ-інцидентами;

– *отримала подальший розвиток* система корелювання подій та управління ІТ-інцидентами, яка за рахунок використання розроблених моделей обробки даних, онтологіко-реляційного сховища даних та інтеграційної шини даних, дає змогу формалізувати інформаційну технологію, що реалізує процеси управління ІТ-інцидентами на ОКІ згідно вимог міжнародних стандартів та найкращих світових практик щодо створення систем управління ІТ-інцидентами.

Практичне значення одержаних результатів. Отримані в дисертаційній роботі результати можуть бути використані в галузі інформаційних технологій для забезпечення стійкого функціонування хмарної інформаційної інфраструктури (у т.ч. критичної) в умовах деструктивних інформаційно-технічних впливів.

Практична цінність роботи полягає у такому:

– на основі дата сету NSL-KDD навчено нейронну мережу з точки зору виявлення аномалій типу DoS, U2R, R2L та Probe.

– створено методику зберігання та класифікації даних, яка дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірностей та забезпечувати високу швидкість пошуку.

– сформовано специфікацію реалізації SIEM-систем на ОКІ, у вигляді основних та додаткових вимог.

– розроблено спеціальний програмний застосунок, який можна використовувати для управління ІТ-інцидентами, які виникають в КІ і мають вплив на критично важливі ресурси (КВР).

– результати дисертації впроваджені і використовуються у діяльності ТОВ «AxxonSoft» (акт про впровадження від 11.03.2026), НДЛ протидії кіберзагрозам авіаційної галузі КАІ (акт про впровадження від 12.02.2024), а

також у навчальному процесі кафедри комп'ютерних наук та кібербезпеки Волинського національного університету імені Лесі Українки для підвищення ефективності підготовки фахівців з ІТ (акт про впровадження від 21.12.2023).

Особистий внесок здобувача. Особистий внесок здобувача. Основні положення і результати дисертаційної роботи, що виносяться до захисту, отримані автором самостійно. У роботах, написаних у співавторстві, автору належать: [14, 16, 18-19] – розроблення структурно-аналітичної моделі оброблення даних в хмарних ІКС; [1, 3, 10, 22] – теоретичне обґрунтування моделі онтологіко-реляційного сховища даних; [1-2, 21] – розроблення моделі інтеграційної шини даних; [1, 9-13, 17, 20] – представлення системи корелювання подій та управління ІТ-інцидентами; [1-3, 5-8, 10-11, 13, 16] – експериментальне дослідження та програмна реалізація запропонованої системи для управління ІТ-інцидентами; [1-2, 4-5, 9-16] – аналіз підходів до управління подіями та ІТ-інцидентами на ОКІ.

З робіт, що опубліковані у співавторстві, у дисертаційній роботі використовуються виключно результати, отримані особисто здобувачем.

Апробація результатів дисертації. Результати досліджень дисертаційної роботи доповідалися та обговорювалися на таких наукових конференціях: «Cybersecurity Providing in Information and Telecommunication Systems» (CPITS) (Kyiv, 2023-2024); «International Conference on Dependable Systems, Services and Technologies» DESSERT-2024 (Athens, Greece 2023); «International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications» IDAACS-2023 (Dortmund, Germany, 2023); «International Conference on Conflict Management in Global Information Networks» CMiGiN (Kyiv, 2022, 2025); «Information Technology and Implementation» IT&I (Kyiv, 2022); «Системи та засоби штучного інтелекту» (Київ, 2021); «АВІА-2023» (Київ, 2023); «Кіберзахист особи, суспільства і держави» (с. Велятино, 2024) та ін.

Публікації. Основні положення дисертації опубліковано у 22 наукових працях, у тому числі: 19 наукових статтях, серед них 11 – у закордонних рецензованих виданнях, які входять до наукометричної бази даних Scopus, 8 – у вітчизняних фахових наукових журналах, а також у 3 матеріалах і тезах доповідей на конференціях.

Структура роботи та її обсяг. Дисертація складається з анотації, вступу, чотирьох розділів, висновків, додатків, списку використаних джерел і має 153 сторінки основного тексту, 90 рисунків, 9 таблиць, 42 сторінки додатків. Список використаних джерел містить 101 найменування і займає 12 сторінок. Загальний обсяг роботи 196 сторінок.

РОЗДІЛ 1
СУЧАСНІ ПІДХОДИ ДО УПРАВЛІННЯ ІТ-ІНЦИДЕНТАМИ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

1.1. Аналіз нормативно-правового забезпечення безпеки на об'єктах критичної інфраструктури держави

Сучасна інформаційна інфраструктура включає значну кількість взаємопов'язаних систем і компонентів, що потребують постійного моніторингу, контролю та захисту. У зв'язку з цим провідні держави світу приділяють значну увагу забезпеченню безпеки та сталого функціонування життєво важливих об'єктів, до яких належать (рис. 1.1) гідротехнічні споруди, енергетичні системи, хімічні виробництва, транспортні вузли, аеродроми тощо.

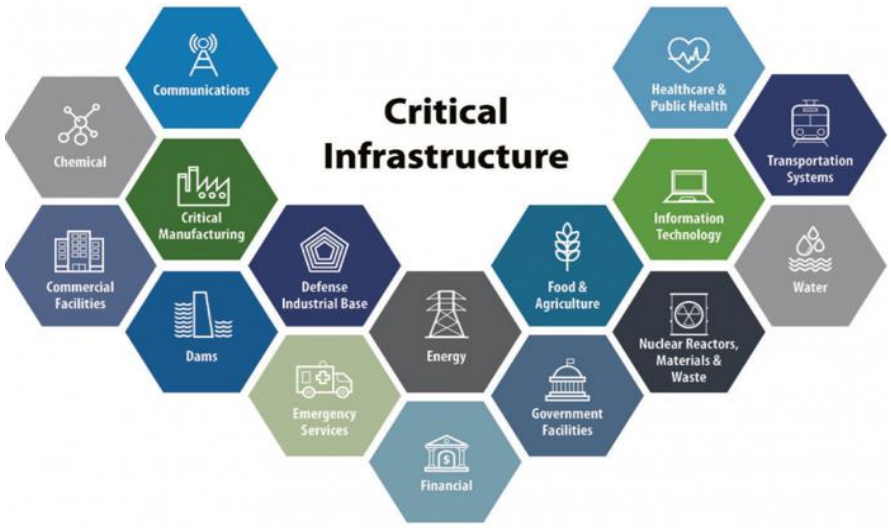


Рис. 1.1. Життєво важливі об'єкти інфраструктури

Відповідно до Закону України «Про критичну інфраструктуру» [1], до життєво важливих функцій або послуг належать ті, що забезпечуються органами державної влади, органами місцевого самоврядування, суб'єктами господарювання та організаціями різних форм власності, а їх порушення або припинення може призвести до суттєвих негативних наслідків для національної безпеки. Таким чином, виведення з ладу відповідних об'єктів може спричинити значні, а в окремих випадках і катастрофічні наслідки для ІКС. Особливої уваги

потребує визначення переліку інформаціних об’єктів, які є критичними, для певної ІКС та способів їх віднесення.

Одне з перших визначень ОКІ, а також необхідність їх систематизації та визначення специфіки віднесення до критичної інфраструктури (КІ), було закріплено в Законі України «Про основні засади забезпечення кібербезпеки України», прийнятому у жовтні 2017 року. Згідно з цим Законом, виокремлюється інформаційна складова КІ – критична інформаційна інфраструктура (КІІ), яка визначається як сукупність об’єктів КІІ (ОКІІ). У свою чергу, під ОКІІ розуміється комунікаційна або технологічна система ОКІ, кібератака на яку може безпосередньо вплинути на стале функціонування такого об’єкта [2]. Крім того, зазначений Закон визначає необхідність ідентифікації ОКІ та формування єдиного реєстру таких об’єктів, а також містить посилання на інші нормативно-правові акти, зокрема [3], який встановлює критерії та порядок віднесення об’єктів до ОКІ, їх перелік і загальні вимоги до кіберзахисту.

Ще одним важливим нормативним документом є постанова Кабінету Міністрів України №1109 «Деякі питання об’єктів критичної інфраструктури» [4], якою затверджено порядок віднесення об'єктів до ОКІ, перелік секторів (підсекторів) та основних послуг КІ держави (рис. 1.2), а також методику категоризації ОКІ. Зазначена методика визначає механізм віднесення ОКІ до відповідної категорії критичності, що здійснюється на основі оцінювання рівня можливого негативного впливу.

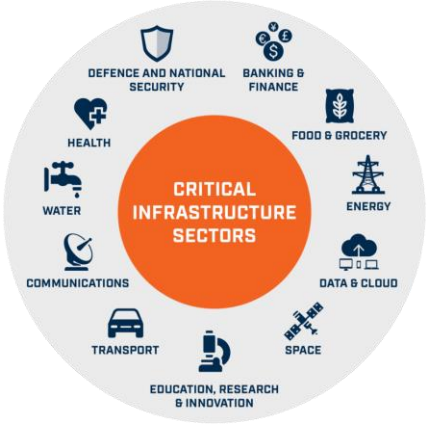


Рис. 1.2. Приклад переліку секторів ОКІ

Але найбільш суттєвим документом у сфері КІ є Закон України «Про критичну інфраструктуру» [1], який набув чинності у листопаді 2021 року та визначає правові й організаційні засади захисту ОКІ під час створення та функціонування національної системи захисту КІ.

Закон визначає базові поняття, зокрема: КІ – це сукупність ОКІ, та відповідно ОКІ – це об'єкти інфраструктури, системи, їх частини та сукупності, які є важливими для економіки, національної безпеки та оборони держави, а порушення їх функціонування може завдати шкоди життєво важливим національним інтересам [1].

Крім того, Закон визначає порядок віднесення об'єктів до КІ, який здійснюється за сукупністю критеріїв, зокрема: виконання функцій із забезпечення життєво важливих національних інтересів; наявність потенційних викликів і загроз щодо об'єктів КІ; ймовірність завдання значної шкоди нормальним умовам життєдіяльності населення; вразливість об'єктів та тяжкість можливих наслідків, що можуть проявлятися у шкоді здоров'ю населення, соціальній сфері, державному суверенітету, економіці та природним ресурсам; масштабність негативних наслідків для держави; тривалість їх ліквідації; а також вплив на функціонування суміжних секторів КІ [1].

Для визначення рівня вимог щодо забезпечення захисту ОКІ відповідно до рівня їх важливості здійснюється їх категоризація за рівнями критичності [1]. Таким чином, нормативно-правовими актами України задекларовано базові аспекти визначення та оцінювання категорій критичності ІКС, проте недостатньо дослідженими залишаються питання забезпечення належного рівня протидії загрозам, а також зниження ризику виникнення аналогічних інцидентів у майбутньому.

На сьогодні в Україні функціонує Урядова команда реагування на комп'ютерні надзвичайні події (CERT/CSIRT) [5], яка здійснює збирання інформації про ІТ-інциденти, їх класифікацію та визначення способів

реагування і нейтралізації. Особливої уваги потребує процес управління та прийняття рішень щодо виявлення та усунення можливих загроз ОКІ.

Серед існуючих інструментів для ефективного реагування на інциденти CSIRT на ОКІ важливе місце займають SIEM-системи [6], функціонування яких полягає в оперативному збиранні, збереженні та аналітичній обробці даних про події безпеки. Такі дані формуються та фіксуються в системних журналах різних апаратних і програмних компонентів інформаційної інфраструктури, зокрема серверів, робочих станцій, маршрутизаторів, мережевих екранів, систем управління базами даних, систем виявлення атак, антивірусних засобів тощо. Основною метою застосування таких систем є підвищення рівня цифрової стійкості та інформаційної безпеки (ІБ) ІКС за рахунок забезпечення можливості оброблення інформації про події безпеки в режимі, близькому до реального часу, а також здійснення кореляції подій і управління ІТ-інцидентами [6].

Питаннями корелювання подій та управління ІТ-інцидентами у т.ч. на ОКІ займаються такі вітчизняні та закордонні вчені: Богачук І. [7], Бурячок В. [7-9], Соколов В. [9], Aslan O. [10], Berdibayev R. [11-13], Karlzen H. [14], Lee J. [15], Pernul G. [16], Vielberth M. [17] та інші. Проте, слід зазначити, що переважна більшість проаналізованих робіт направлена на вирішення однієї чи двох задач захисту ОКІ. А для успішної реалізації заходів захисту ОКІ необхідне вирішення низки завдань, основне з яких пов'язане зі створенням єдиної системи моніторингу загроз безпеці, головною метою якої буде зниження до мінімального рівня ризику впливу на ОКІ і мінімізація збитків, що можуть виникнути внаслідок реалізації загроз.

1.2. Аналіз існуючих підходів до оброблення даних у хмарних системах критичної інфраструктури

Сучасні технології виявлення шкідливого програмного забезпечення (ШПЗ) включають складні математичні методи, а також апаратно-програмні комплекси для зберігання, оброблення та передачі даних,

комп’ютеризованого управління і телекомунікацій. Постійний розвиток обчислювальної техніки та засобів автоматизації, а також зростання попиту на хмарні системи виявлення ШПЗ (рис. 1.3) [18] зумовлюють збільшення обсягів метаданих, що передаються до таких систем.

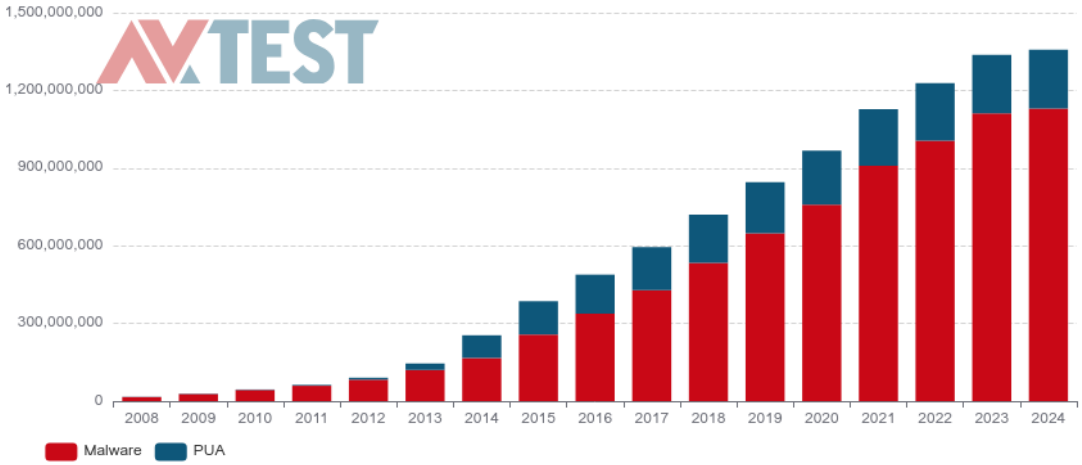


Рис. 1.3. Динаміка ШПЗ та потенційно небажаних програм (PUA) у світі (за даними AV-TEST [18])

Таким чином, постає проблема розроблення математичних моделей, які максимально точно формалізують технологію функціонування ІКТ. Особливо важливим є завдання математичного опису технології хмарних систем виявлення ШПЗ в ІКТ з урахуванням зазначених факторів (неоднорідність, багатозв’язність тощо).

На сьогодні хмарні системи є однією з найперспективніших технологій зберігання інформації та ефективного надання послуг у мережі. Використання цієї технології для захисту комп’ютерних систем від кібератак має низку переваг порівняно з традиційними підходами, зокрема простоту використання, доступність, нижчу вартість і масштабованість. Під терміном ШПЗ будемо розуміти будь-яке програмне забезпечення, призначене для несанкціонованого впливу на комп’ютерні системи з метою порушення їх функціонування або завдання шкоди кінцевим пристроям.

Відповідно до [10], захищеними активами можуть бути настільні та портативні комп’ютери, комп’ютерні системи і мережі, мобільні пристрої,

пристрої Інтернету речей (IoT), кіберфізичні системи (CPS), а також ОКІІ держави, у яких можуть виникати різні види аномалій.

Отже, детально розглянемо низку досліджень, присвячених виявленню аномалій у хмарному середовищі.

У роботі [19] представлено підхід до захисту пристроїв IoT від атак з боку локальних комп'ютерних мереж. Авторами запропоновано концепцію глибокого навчання на основі поведінки (BDLF), яка інтегрована у хмарну платформу середовища IoT. У межах цього підходу спочатку на основі аналізу викликів API будуються графи поведінки, після чого за допомогою нейронних мереж типу Stacked AutoEncoders (SAE) здійснюється вилучення високорівневих характеристик таких графів. Результати експериментів демонструють, що запропонований підхід дозволяє виявляти різноманітні види аномалій та підвищує середню точність виявлення на 1,5 %.

У роботі [20] запропоновано метод виявлення аномалій на основі аналізу мережного трафіку та поведінки програмного забезпечення в комп'ютерних системах (КС). Метод ґрунтується на класифікації множин API-викликів, їх частот та послідовностей виконання, отриманих із графів потоків керування програмних додатків, а також ознак, вилучених із DNS-трафіку мережі. Результати експериментів показали достовірність виявлення різних видів аномалій на рівні від 97,29 до 99,42 %, що підтверджує ефективність підходу для підвищення точності виявлення ШПЗ у КС.

У дослідженні [21] автори запропонували енергоефективну модель хостингу, що базується на використанні компонентів хмарних сервісів Amazon для підвищення масштабованості та ефективності системи. У роботі проведено порівняльний аналіз із традиційними антивірусними рішеннями. Отримані результати свідчать, що запропонований підхід забезпечує кращу продуктивність порівняно з класичними антивірусами. Водночас інфраструктуру виявлення аномалій можна вдосконалити шляхом інтеграції механізмів виявлення вторгнень, що підтримуються хмарним середовищем.

У роботі [22] запропоновано хмарну систему виявлення та придушення різного виду аномалій для бездротових мультимедійних систем IoT на основі динамічної диференціальної гри. У запропонованій моделі реалізовано виявлення аномалій на основі методу опорних векторів (SVM) із використанням хмарної платформи безпеки. Кількість заражених вузлів визначається відповідно до характеристик бездротової мультимедійної системи, а перехід станів описується модифікованою епідемічною моделлю. Результати дослідження показали ефективність запропонованого підходу для систем з обмеженими ресурсами.

У дослідженні [23] запропоновано інформаційну технологію виявлення аномалій у хмарному середовищі на основі методів машинного навчання (ML). Авторами використано підхід мінімізації логарифмічних втрат із застосуванням моделей, зокрема KNN та логістичної регресії. На основі отриманих результатів визначено найбільш ефективну модель, яку було реалізовано у вигляді хмарного сервісу на платформі AWS. Запропонований підхід демонструє ефективність у задачах визначення легітимності файлів, однак може бути вдосконалений шляхом застосування методів відбору ознак та більш складних моделей навчання.

У роботі [24] представлено хмарне рішення для виявлення аномалій під назвою TrustAV, яке базується на методі зіставлення шаблонів. Обробка даних і аналіз ШПЗ здійснюються на віддаленому сервері, що дозволяє використовувати систему як хмарний сервіс. Для забезпечення безпеки обробки даних застосовуються технології Intel SGX. Разом із тим, у роботі відсутні результати експериментальної перевірки на реальних даних.

Для виявлення зростаючої кількості атак ШПЗ у роботі [10] запропоновано інтелектуальну систему виявлення аномалій на основі поведінкового аналізу в хмарному середовищі. Запропонована система формує набір даних про аномалії на різних віртуальних машинах, що дозволяє ефективно визначати характерні ознаки атак (рис. 1.4). Після цього

відібрані на основі навчання та правил зразки передаються агентам виявлення для відокремлення аномалій від безпечних зразків.

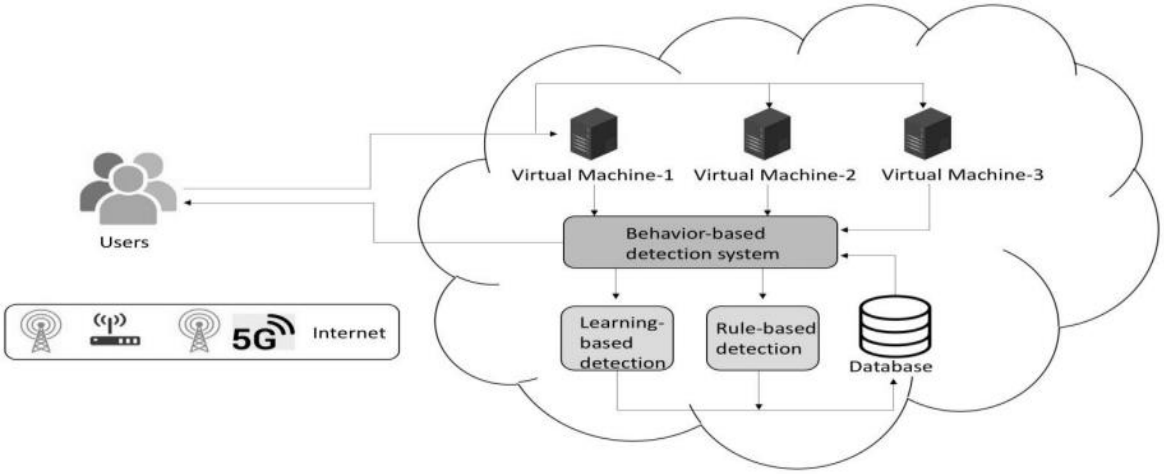


Рис. 1.4. Хмарна архітектура виявлення ШПЗ

У табл. 1.1, відповідно до [18, 25-28], систематизовано результати аналізу підходів до виявлення аномалій у хмарному середовищі за такими критеріями: 1) Ясність формалізації; 2) Гнучкість та універсальність; 3) Можливість самонавчання та удосконалення; 4) Точність виявлення; 5) Експериментальне підтвердження.

Таблиця 1.1

Аналіз підходів до виявлення аномалій у хмарному середовищі

№	Назва підходу	Критерії				
		1	2	3	4	5
1	Підхід виявлення аномалій на основі глибокого вивчення графіків поведінки	+	-	+	+	+
2	Метод виявлення аномалій шляхом аналізу мережного трафіку та поведінки ПЗ в КС	-	-	+	+	+
3	Хмарна енергоефективна модель хостингу для виявлення аномалій	-	+	+	-	-
4	Модель виявлення та придушення аномалій для бездротової системи IoT на основі динамічної диференціальної гри	+	+	+	-	+
5	Інформаційна технологія виявлення аномалій в хмарному середовищі на основі ML	+	+	+	-	-
6	Модель виявлення аномалій у хмарному середовищі: TrustAV	+	+	+	-	-
7	Система виявлення аномалій на основі аналізу поведінки в хмарному середовищі	+	-	+	+	+

Загальна мета всіх цих досліджень полягає в тому, щоб ідентифікувати аномалії шляхом збільшення швидкості виявлення при одночасному зниженні рівня неправильної класифікації. Вивчаючи ці дослідження, можна побачити, що, хоча кожен метод виявлення має свої переваги та працює краще для певних наборів даних у хмарному середовищі, жоден із них не може виявити всі сто відсотків ШПЗ. Отже, існує необхідність створення є створення й дослідження нової моделі інформаційної технології ефективного оброблення даних у хмарних системах виявлення аномалій на ОКІ.

1.3. Аналіз існуючих сховищ даних для об'єктів критичної інфраструктури

У сучасному світі кількість кіберзагроз безперервно зростає, і одним із ефективних засобів захисту інформації є використання SIEM-систем. В основі їх функціонування лежить застосування баз даних. База даних – це організована сукупність структурованих даних, що зберігаються у цифровому вигляді в КС. Управління базами даних здійснюється за допомогою систем управління базами даних (СУБД). Дані разом із СУБД та прикладними програмами утворюють систему баз даних. Сучасні бази даних зазвичай зберігають інформацію у вигляді таблиць, де дані представлені у формі рядків і стовпців. Така організація забезпечує зручність керування даними, зокрема їх додавання, редагування, видалення, оновлення та контроль. Більшість баз даних використовують мову структурованих запитів SQL для внесення та отримання інформації.

Наразі існує широке різноманіття типів баз даних [14, 16, 29-31, 62]. Вибір відповідного типу бази даних для конкретної SIEM-системи визначається особливостями використання даних у відповідному контексті. Під терміном «типи баз даних» розуміють шаблони та структури, що застосовуються для організації інформації в СУБД [29-31].

Водночас для забезпечення семантичної узгодженості та інтеграції даних у складних інформаційних системах (ІС) доцільним є використання

онтології як формалізованої моделі предметної області, що дозволяє визначити сутності, їх властивості та взаємозв'язки між ними. Розглянемо наступні типи баз даних, які застосовуються в SIEM-системах згідно з [44, 59, 62]:

1. Найпростіші типи баз даних

Спочатку доцільно розглянути базові типи баз даних, які й досі використовуються в спеціалізованих середовищах, проте значною мірою витіснені більш надійними та ефективними рішеннями.

1.1. Прості структури даних

Найпростішим способом зберігання даних є використання текстових файлів. Такий підхід підходить для роботи з невеликими обсягами інформації. Для розділення полів використовуються спеціальні символи: кома або крапка з комою у CSV-файлах, двокрапка або пробіл у *nix-подібних системах:

root:x:0:0:root:/root:/bin/bash

bin:x:2:2:bin:/bin:/usr/sbin/nologin

sys:x:3:3:sys:/dev:/usr/sbin/nologin

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

1.2. Ієрархічні бази даних

На відміну від текстових структур, у цьому типі баз даних встановлюються зв'язки між об'єктами. В ієрархічних базах даних кожен запис має лише одного предка, що формує деревоподібну структуру. Записи класифікуються відповідно до їхнього розташування в ієрархії. Узагальнену структуру ієрархічних баз даних наведено на рис. 1.5 [44, 62]

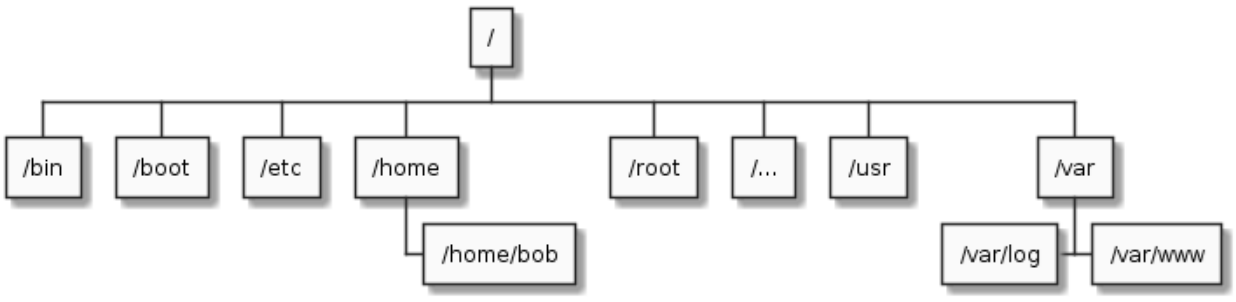


Рис. 1.5. Структура ієрархічних баз даних

1.3. Мережеві бази даних

Мережеві бази даних розширюють можливості ієрархічної моделі: записи можуть мати більше одного предка, що дозволяє моделювати складні взаємозв'язки між даними. Узагальнену структуру мережевих баз даних відповідно до [44] наведено на рис. 1.6.

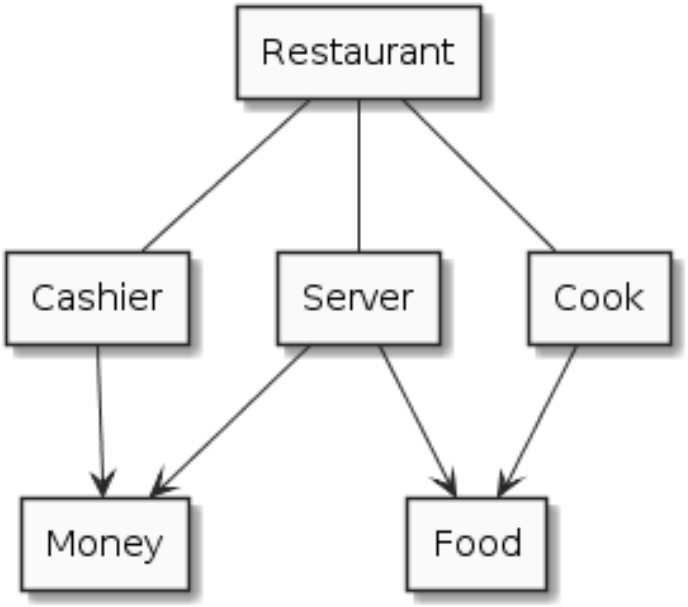


Рис. 1.6. Структура баз даних мережі

2. Реляційні бази даних

2.1. Бази даних SQL

Реляційні бази даних є одним із найстаріших і водночас найбільш поширених типів баз даних загального призначення. Дані в реляційних базах даних організовані у вигляді таблиць, що складаються зі стовпців і рядків. Кожен стовпець має власну назву та визначений тип даних. Кожен рядок представляє окремий запис, який містить значення для всіх відповідних стовпців таблиці. Узагальнену структуру реляційних баз даних наведено на рис. 1.7.

2.2. OLTP-бази даних

OLTP-база даних – це тип бази даних, призначений для оброблення транзакцій у режимі реального часу, які виконуються одночасно великою кількістю користувачів. Узагальнену структуру OLTP-баз даних наведено на рис. 1.8 [62].

Реляційні бази даних широко застосовуються в SIEM-системах, зокрема в IBM QRadar, LogRhythm, AlienVault USM, AlienVault OSSIM, Splunk, FortiSIEM, Wazuh, SolarWinds, ManageEngine, Prelude OSS, Prelude SIEM, Sagan, EventTracker, Trustwave SIEM Enterprise, McAfee ESM [15, 33-34].

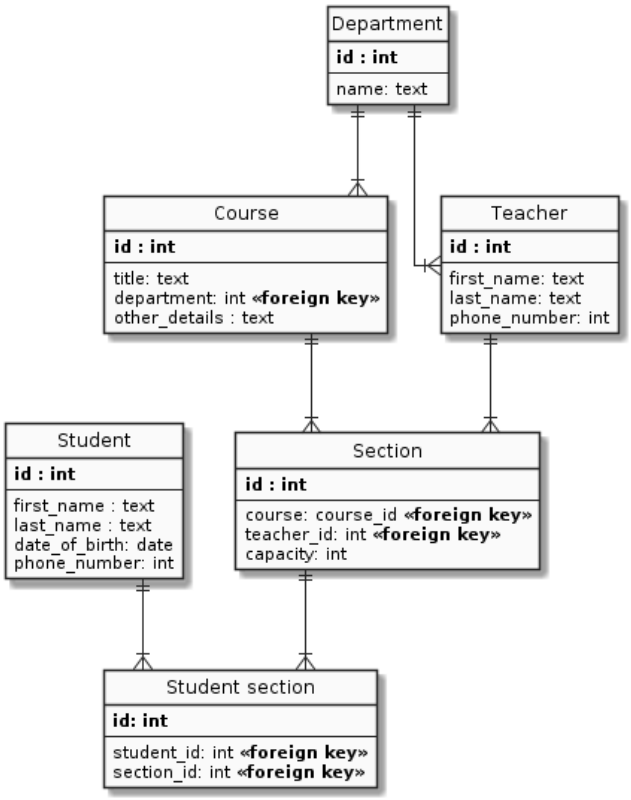


Рис. 1.7. Структура реляційних баз даних

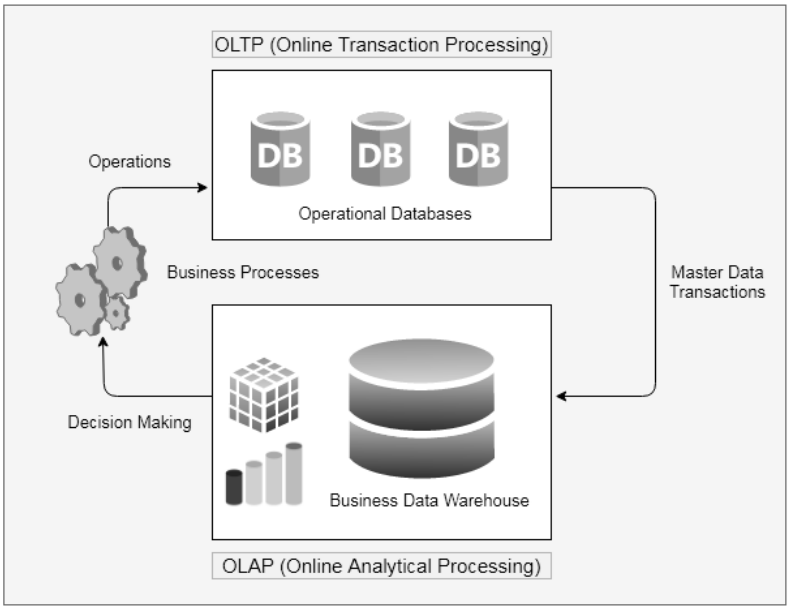


Рис. 1.8. Структура OLTP-баз даних

3. Бази даних NoSQL

NoSQL – це група типів баз даних, які реалізують підходи, відмінні від традиційної реляційної моделі. Термін NoSQL означає «не лише SQL» і підкреслює можливість використання альтернативних або SQL-подібних механізмів запитів.

Бази даних NoSQL (нереляційні бази даних) забезпечують можливість зберігання та оброблення неструктурованих або напівструктурованих даних, на відміну від реляційних баз даних, які передбачають жорстко визначену схему [44, 62]. Популярність NoSQL-баз даних зростає у зв'язку зі збільшенням складності веб-додатків та обсягів оброблюваних даних .

3.1. Бази даних типу «ключ-значення»

У базах даних типу «ключ–значення» для зберігання інформації використовується пара: унікальний ключ та відповідне йому значення (інформаційний об'єкт даних). Як значення можуть виступати, наприклад, JSON-об'єкти, зображення або текстові дані. Для отримання даних система здійснює пошук за ключем і повертає відповідне значення (blob). Узагальнену структуру NoSQL-баз даних типу «ключ–значення» наведено на рис. 1.9.

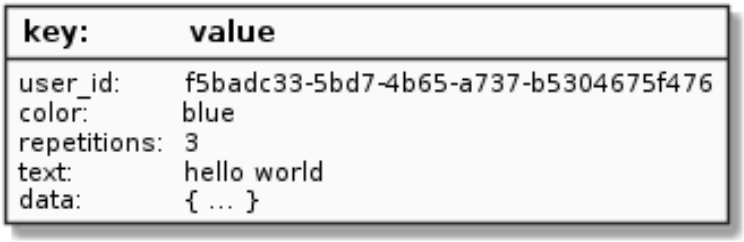


Рис. 1.9. Структура баз даних типу «ключ-значення»

3.2. Документо-орієнтовані бази даних

Документо-орієнтовані бази даних (також відомі як бази даних документів або сховища документів) поділяють базову семантику доступу та пошуку зі сховищами типу «ключ–значення». Такі бази даних також використовують ключ для унікальної ідентифікації даних. Основна відмінність між сховищами типу «ключ–значення» та документо-орієнтованими базами даних полягає в тому, що замість зберігання неструктурованих блоків даних

останні зберігають інформацію у структурованих форматах, таких як JSON, BSON або XML. Узагальнену структуру документо-орієнтованих баз даних наведено на рис. 1.10 [34-35, 44, 62].

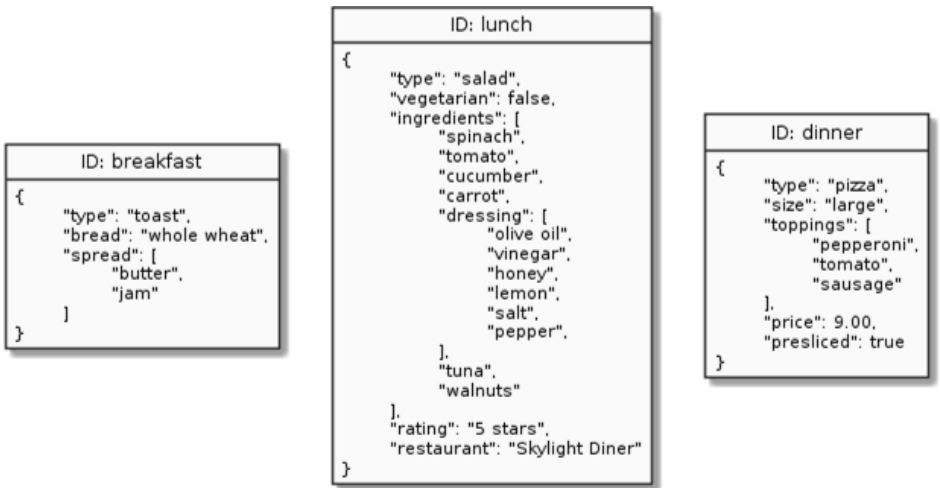


Рис. 1.10. Структура документо-орієнтованих бази даних

3.3. Графові бази даних

Замість відображення зв'язків за допомогою таблиць і зовнішніх ключів графові бази даних встановлюють зв'язки між об'єктами за допомогою вузлів, ребер та їх властивостей. Узагальнену структуру графових баз даних наведено на рис. 1.11.

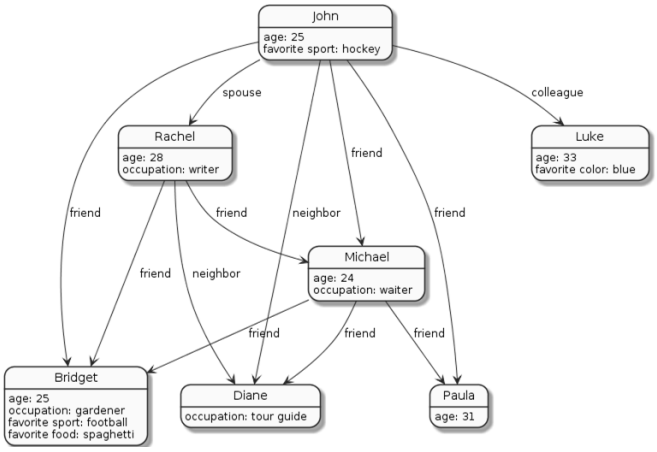


Рис. 1.11. Структура графових баз даних

Графові бази даних представляють інформацію у вигляді вузлів, кожен з яких може мати довільну кількість властивостей. Дані в таких базах зберігаються у вигляді сутностей і зв'язків між ними.

3.4. Стовпчикові бази даних

Стовпчикові (колонкові) бази даних належать до категорії NoSQL-систем і також відомі як сховища з широкими стовпцями. Незважаючи на це, за зовнішньою структурою вони частково подібні до реляційних баз даних. Узагальнену структуру стовпчикових баз даних наведено на рис. 1.12 [44].

Подібно до реляційних баз даних, стовпчикові бази даних зберігають інформацію у вигляді рядків і стовпців, однак відрізняються принципом організації та взаємозв'язків між елементами.

У реляційних базах даних усі рядки відповідають фіксованій схемі, яка визначає набір стовпців, їх типи даних та інші характеристики. У стовпчикових базах даних, натомість, використовуються структури, відомі як «сім'ї стовпців», замість традиційних таблиць. Сім'ї стовпців містять рядки, кожен із яких може мати власний формат. Кожен рядок складається з унікального ідентифікатора, що використовується для пошуку, після чого задаються набори імен стовпців і відповідних їм значень.

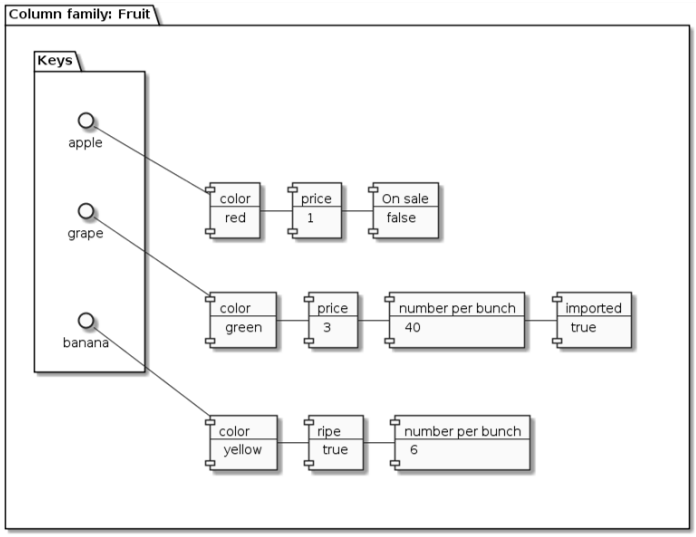


Рис. 1.12. Структура стовпчикових баз даних

3.5. Бази даних часових рядів

Бази даних часових рядів призначені для збирання та оброблення даних, що змінюються з часом. Такі бази даних зазвичай організовані у вигляді структур, у яких фіксуються значення певних параметрів у часі [62]. Наприклад,

може бути створена таблиця для відстеження температури процесора, де кожен запис містить позначку часу та відповідне значення температури. У межах однієї структури може зберігатися декілька метрик. Узагальнену структуру баз даних часових рядів наведено на рис. 1.13.

Time	CPU Temp	System Load	Memory Usage %
2019-10-31T03:48:05+00:00	37	0.85	92
2019-10-31T03:48:10+00:00	42	0.87	90
2019-10-31T03:48:15+00:00	33	0.74	87
2019-10-31T03:48:20+00:00	34	0.72	77
2019-10-31T03:48:25+00:00	40	0.88	81
2019-10-31T03:48:30+00:00	42	0.89	82
2019-10-31T03:48:35+00:00	41	0.88	82

Рис. 1.13. Структура баз даних часових рядів

NoSQL бази даних використовують такі SIEM-системи: AlienVault USM, AlienVault OSSIM, MozDef.

4. Комбіновані бази даних

NewSQL та багатомодельні бази даних є різними типами систем зберігання даних, проте вони спрямовані на розв'язання спільного кола проблем, що виникають унаслідок використання відмінних підходів SQL і NoSQL. Поєднання переваг цих підходів дозволяє підвищити ефективність оброблення даних у сучасних інформаційних системах.

4.1. Бази даних NewSQL

Бази даних NewSQL поєднують реляційну модель даних і семантику SQL із сучасними масштабованими архітектурними рішеннями. Їх основною метою є забезпечення більшої масштабованості порівняно з традиційними реляційними базами даних, а також вищого рівня узгодженості, ніж у NoSQL-системах. Компроміс між узгодженістю, доступністю та стійкістю до розподілення є фундаментальною проблемою розподілених систем і описується теоремою CAP [36-37, 44, 62].

4.2. Багатомодельні бази даних

Багатомодельні бази даних поєднують функціональні можливості кількох типів баз даних у межах однієї системи. Такий підхід дозволяє використовувати різні моделі представлення даних залежно від їх структури та

призначення. Інтеграція різних моделей даних у межах однієї системи забезпечує можливість виконання операцій, які в іншому випадку були б складними або неможливими. Зокрема, багатомодельні бази даних дозволяють здійснювати доступ і керування даними, що зберігаються в різних форматах, у межах одного запиту, а також підтримувати узгодженість при виконанні змін у декількох підсистемах одночасно [38].

5. Об'єктно-орієнтовані бази даних (ООБД)

В ООБД інформація представлена у вигляді об'єктів, аналогічно до підходів об'єктно-орієнтованого програмування. Узагальнену структуру ООБД наведено на рис. 1.14.

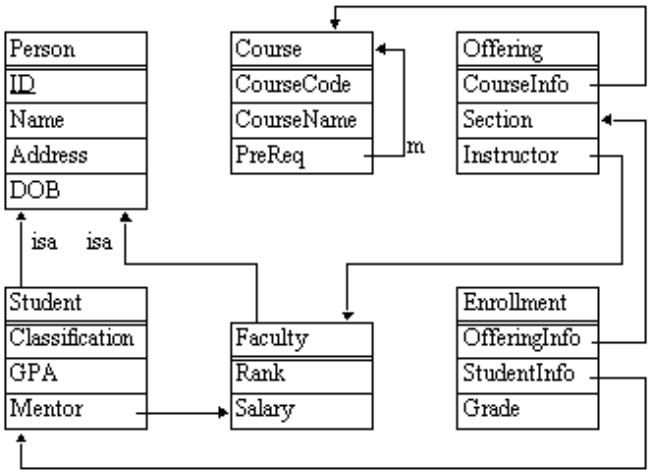


Рис. 1.14. Структура баз даних ООБД

6. Хмарні бази даних

Хмарна база даних – це сукупність структурованих або неструктурованих даних, розміщених на приватній, публічній або гібридній платформі хмарних обчислень [11, 15, 39-40]. Існують дві основні моделі хмарних баз даних: традиційна модель та база даних як послуга (Database as a Service, DBaaS). У моделі DBaaS адміністративні функції та обслуговування виконує постачальник хмарних сервісів. Узагальнену структуру хмарних баз даних наведено на рис. 1.15 [41-44, 62].

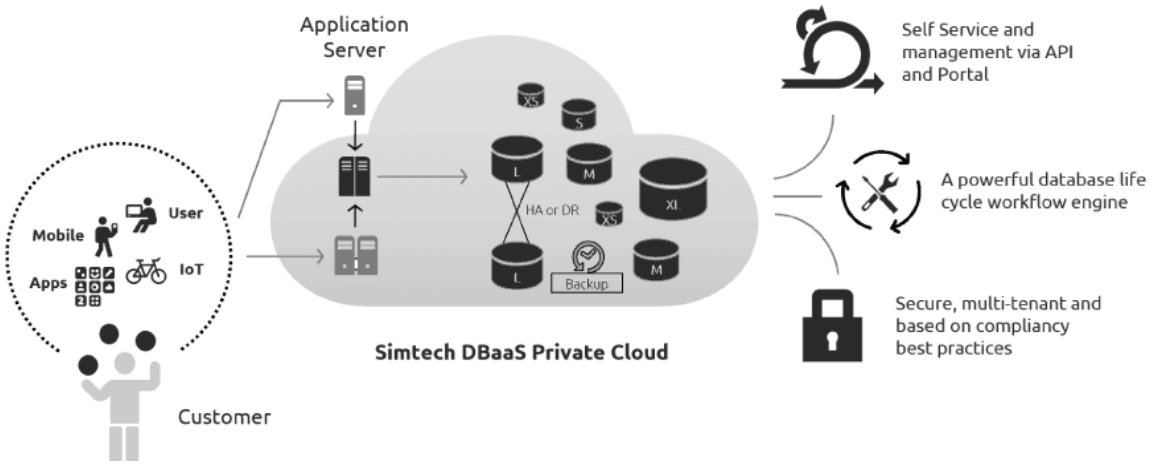


Рис. 1.15. Структура хмарних баз даних

Хмарні типи баз даних використовують такі SIEM-системи: HPE ArcSight, Splunk, Micro Focus ArcSight, Trustwave SIEM Enterprise.

Результати аналізу СУБД в різних SIEM та згідно до [43-44, 59] наведено в табл. 1.2.

Таблиця 1.2

Порівняльний аналіз вживаних СУБД для різних SIEM-систем

SIEM	**СУБД**
IBM QRadar	Ariel database, PostgreSQL, SQLite
LogRhythm	Oracle, SQL Server, MySQL
Splunk	DB2 / Linux, Informix, MemSQL, MySQL, AWS Aurora, Microsoft SQL Server, Oracle, PostgreSQL, AWS RedShift, SAP SQL Anywhere, Sybase ASE, Sybase IQ, and Teradata
McAfee (ESM)	MSSQL, Oracle, MySQL, Data Access Server (DAS), DB2 / UDB
AlienVault USM	RedisDB, MySQL
AlienVault OSSIM	RedisDB, MySQL
FortiSIEM	PostgreSQL
MozDef	RabbitMQ, MongoDB, Elasticsearch, Kibana
Wazuh	MySQL, PostgreSQL
Prelude OSS	MySQL, PostgreSQL
Prelude SIEM	MySQL, PostgreSQL
Sagan	MySQL, PostgreSQL
SolarWinds	MSSQL, Oracle, MySQL, MariaDB.
ManageEngine	Oracle, SQL, DB2, MySQL
EventTracker	Microsoft SQL Server
Micro Focus ArcSight	Own development CORR-E
Trustwave SIEM Enterprise	Microsoft SQL Server, Microsoft SQL Azure, ORACLE, SYBASE, MySQL, IBM, DB2, Hadoop

Після проведеного аналізу (табл. 1.2) та відповідно до [44, 62] слід зазначити, що кожен із розглянутих типів СУБД залишається актуальним у відповідних умовах застосування, де взаємозв'язки між даними визначаються структурою бази даних. Крім того, доцільно розглядати можливість використання гібридних баз даних, які поєднують різні підходи, зокрема SQL та NoSQL. Це дозволяє забезпечити ефективне зберігання, класифікацію та високошвидкісну обробку великих обсягів даних.

Для забезпечення сталого функціонування сучасна SIEM-система повинна оперативно вирішувати комплекс завдань. З одного боку, необхідно забезпечити високошвидкісне зберігання, оброблення та пошук подій у журналах. З іншого боку, важливим є надійне та структуроване зберігання службових даних користувачів, метаданих, конфігураційних параметрів, а також архівів подій безпеки [59].

Використання однієї бази даних для реалізації всіх зазначених функцій не відповідає вимогам сучасної архітектури та ІБ.

У зв'язку з цим виникає необхідність розроблення моделі гібридного онтологіко-реляційного сховища даних, що базується на інтегрованому використанні декількох типів баз даних із різними характеристиками.

1.4. Аналіз існуючих шин даних для ефективного функціонування системи управління IT-інцидентами

Особливості сучасних ESB або інтеграційних шин даних

Інтеграційна шина даних (ІШД) – це програмне забезпечення, яке забезпечує централізований обмін повідомленнями між інформаційними системами та прикладними програмами [12, 45-46].

Сучасні ІШД характеризуються низкою *функціональних можливостей і особливостей*: 1. Забезпечення централізованої інтеграції різнорідних інформаційних систем. 2. Підтримка збору даних із різних джерел (внутрішніх і зовнішніх) у вихідному форматі. 3. Можливість перетворення

даних у необхідні формати для подальшої передачі. 4. Реалізація маршрутизації повідомлень на основі заданої логіки. 5. Наявність механізмів журналювання подій для виявлення помилок та відновлення даних. 6. Підтримка брокерів повідомлень для забезпечення асинхронної взаємодії. 7. Забезпечення моніторингу та контролю інтеграційних процесів.

Разом з тим, використання ІШД *супроводжується певними обмеженнями*: 1. Зміна або модернізація окремих компонентів може вимагати значної переробки інтеграційних інтерфейсів. 2. Реалізація логування може відрізнятися в різних інтеграціях, що ускладнює аналіз подій. 3. Додавання нових компонентів системи може потребувати значних ресурсів на інтеграцію. 4. Ускладнення бізнес-аналітики через різноманітність форматів і джерел даних. 5. Збільшення інфраструктури призводить до зростання витрат на її підтримку.

Порівняльний аналіз ІШД

Нижче розглянемо, як компоненти ІШД реалізовані в рішеннях, які найчастіше пропонуються на казахстанському ринку (Talend [54], Mule [55], WSO2 [56], Red Hat Fuse) (табл. 1.3).

Таблиця 1.3

Порівняльний аналіз ІШД

№	Критерій / ІШД	Talend	Mule	WSO2	Red Hat Fuse
1	Наявність студії	+	+	+	±
2	Підтримка брокера повідомлень	JMS 1.1, Microsoft MQ 3.0, JBoss Messaging 1.4.4, IBM MQ 8.0, Apache ActiveMQ 5.13.2	Anypoint MQ, IBM MQ, Apache Kafka, JMS 1.0.2, 1.1, 2.0 support	Amazon SQS, JMS support, Apache Kafka	Apache ActiveMQ, Apache Kafka, AWS MQ, RabbitMQ, JMS support
3	Логування	статистика виконання завдань і	логування в межах кожної інтеграції,	Логування на основі Apache Log4j за	Логування на основі Apache Log4j через

		компонентів, помилок, попереджень і винятків на рівні завдань, потоків даних всередині завдань; логування в Elastic, Apache Log4j, Apache Commons Logging, Trace Logs	створеної в Mule: помилки та події, обов'язкові для логування за логікою інтеграції; логування запуску, зупинки, розгортання та відключення сервісів та інтеграцій Mule	допомогою бібліотеки Apache Commons Logging. Події системи та компонентів логеруються окремо	бібліотеку Apache Commons Logging, SLF4J, java.util.logging, Elastic
4	Моніторинг	+	+	+	+

Розробники також розглядають такі рішення, як Apache Kafka, Kafka Connect та RabbitMQ, однак ці системи не належать до класичних ІШД, а виконують функції брокерів повідомлень, тому їх включення до даного аналізу є недоцільним. У якості критеріїв оцінювання обрано основні функціональні компоненти ІШД: наявність середовища розробки (студії), підтримка брокерів повідомлень, механізми логування та засоби моніторингу.

1.5. Аналіз сучасних технологій та систем корелювання подій та управління IT-інцидентами на об’єктах критичної інфраструктури

На сьогодні управління інформацією та подіями ІБ (SIEM, Security Information and Event Management) є одним із ключових напрямів розвитку засобів захисту, що забезпечує ефективне виявлення загроз і формування відповідних заходів протидії для підтримки рішень необхідного рівня захисту інформаційної інфраструктури. Для вирішення завдань, пов’язаних із забезпеченням безпеки та фіксацією подій, у роботах [12-13, 43, 60-61] розглянуто основні характеристики функціонування існуючих SIEM-систем та проведено їх порівняльний аналіз. Розглянемо деякі з них більш детально.

Аналіз існуючих SIEM-систем

1.1. IBM QRadar Security Intelligence Platform [31] складається з низки інтегрованих систем збору подій, моніторингу, аналізу захищеності та розслідування інцидентів (рис. 1.16).

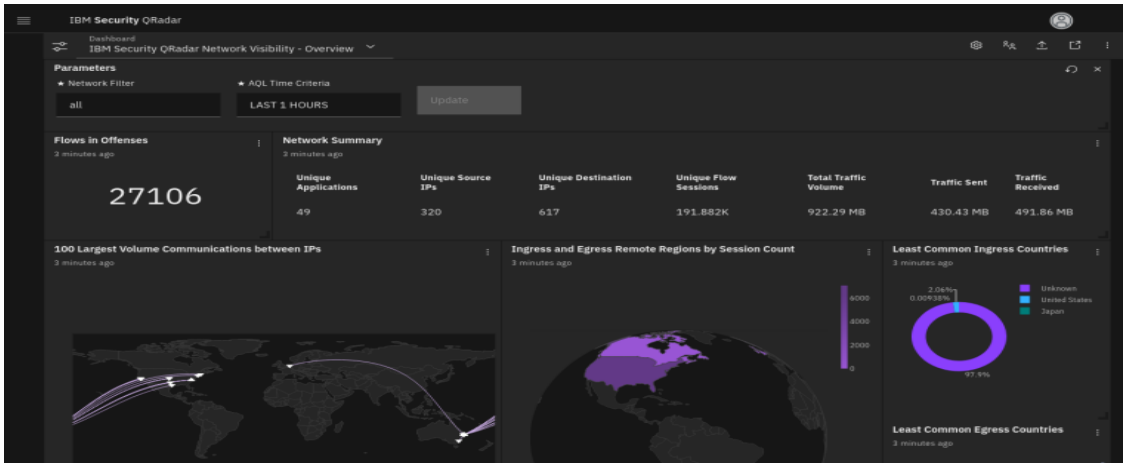


Рис. 1.16. Графічний інтерфейс IBM QRadar

Переваги IBM QRadar: єдина платформа для побудови центрів моніторингу безпеки (SOC), що забезпечує збирання та аналіз подій ІБ, виявлення аномальної мережевої активності, виявлення вразливостей і аналіз конфігурацій; інтеграція з технологіями штучного інтелекту (IBM Watson), засобами мережевої форензики та системами реагування на інциденти (IBM Resilient); гнучка архітектура QRadar Platform, що дозволяє адаптувати ролі та функції модулів відповідно до потреб користувача; наявність широкого набору безкоштовних додатків, контенту та інтеграційних модулів [13, 43].

1.2. LogRhythm [16] – це платформа управління подіями ІБ, яка забезпечує інтелектуальний аналіз журналів і мережевого трафіку в операційних системах Windows і Linux із використанням технологій штучного інтелекту (рис. 1.17).

*Переваги системи:*наявність масштабованого сховища даних; фективність у середовищах із відсутністю централізованого управління та структурованих даних; придатність для використання в малих і середніх організаціях; можливість фільтрації надлишкових даних і концентрації аналізу

на мережевому рівні; сумісність із широким спектром джерел журналів і пристроїв, а також можливість інтеграції з іншими системами (зокрема Varonis) для розширення функцій реагування на інциденти.

Рис. 1.17. Графічний інтерфейс адміністратора LogRhythm

1.3. Splunk – це платформа для аналізу даних, яка використовує технології штучного інтелекту та машинного навчання для отримання практичних, ефективних і прогнозних аналітичних результатів (рис. 1.18). Splunk [14-16, 29] підходить для організацій різного масштабу та підтримує як локальне розгортання, так і використання у вигляді хмарного сервісу (SaaS).

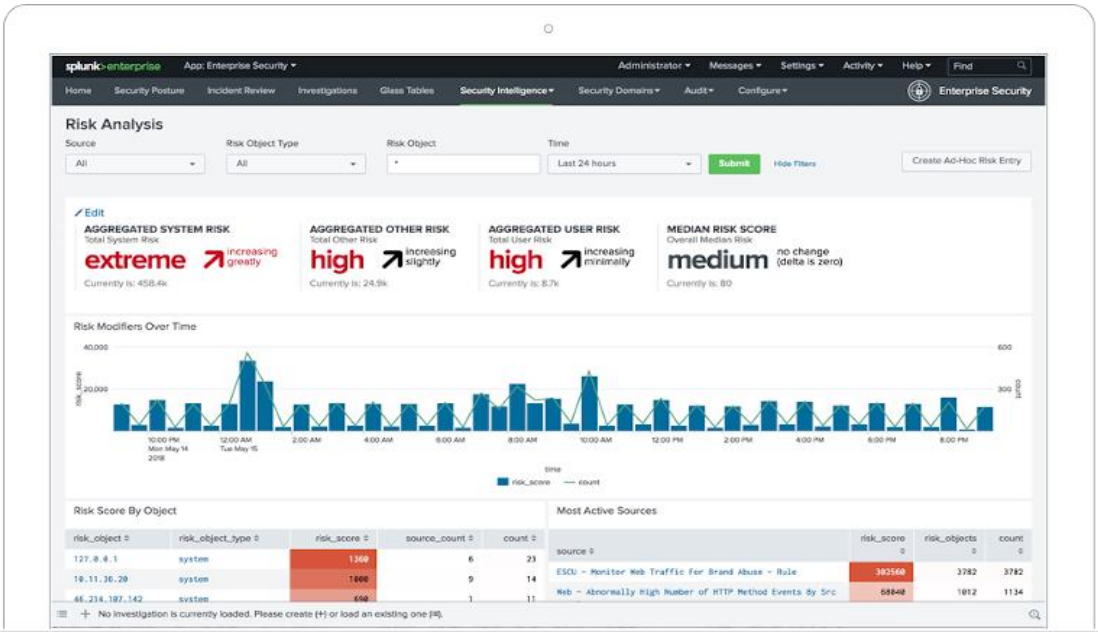


Рис. 1.18. Графічний інтерфейс адміністратора Splunk

Ключові переваги: швидке виявлення загроз; оцінювання та аналіз ризиків; управління сповіщеннями та подіями; структуризація та впорядкування подій безпеки; оперативне та ефективне реагування на інциденти; підтримка роботи з даними як у локальному середовищі, так і в хмарній інфраструктурі [13].

1.4. McAfee Enterprise Security Manager (ESM) [16, 60] є SIEM-платформою, яка може постачатися у вигляді апаратного або програмного рішення, а також у віртуалізованому середовищі. До її складу входять три основні компоненти: ESM, Event Receiver та Enterprise Log Manager, які можуть розгортатися як єдина система або окремо для використання в розподілених чи великомасштабних інфраструктурах (рис. 1.19).

Рис. 1.19. Графічний інтерфейс адміністратора McAfee ESM

Переваги McAfee ESM [13]: підтримка рішень інтеграції з промисловими системами управління (ICS) та системами диспетчерського управління і збору даних (SCADA); використання технології McAfee Data Exchange Layer (DXL), яка забезпечує інтеграцію зі сторонніми рішеннями без необхідності прямої взаємодії через API; можливість використання платформи як централізованого середовища управління подіями безпеки; інтеграція з McAfee Global Threat Intelligence, що забезпечує доступ до актуальної інформації про загрози та дозволяє оперативно виявляти підозрілі мережеві з'єднання, зокрема взаємодію з потенційно шкідливими IP-адресами.

1.5. AlienVault USM – це багатофункціональна платформа управління ІБ, яка забезпечує централізацію процесів виявлення загроз, реагування на інциденти та контролю відповідності стандартам у хмарних і локальних середовищах (рис. 1.20).

Ключові можливості [29, 43]: збір та аналіз даних безпеки з різних сторонніх джерел; візуалізація даних у вигляді інформаційних панелей (dashboards) із розширеними аналітичними можливостями; управління діями інтегрованих засобів безпеки на основі отриманої аналітичної інформації; розширення функціональності системи за рахунок підключення додаткових модулів (AlienApps).

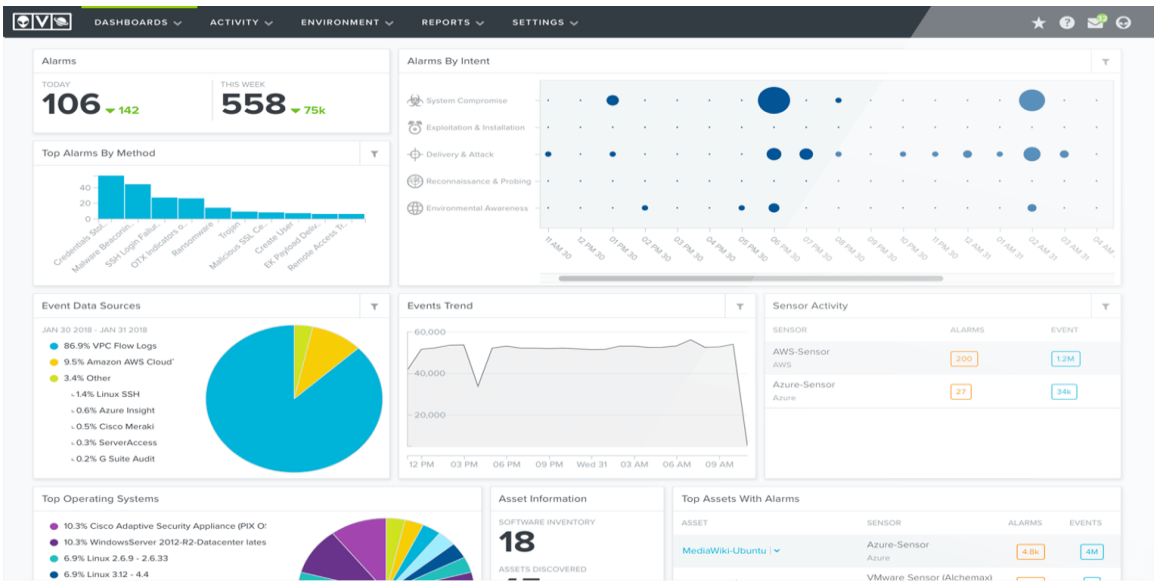


Рис. 1.20. Інтерфейс AlienVault USM

1.6. FortiSIEM – це комплексна масштабована платформа управління безпекою, продуктивністю та відповідністю вимогам для всіх компонентів інформаційної інфраструктури, яка підтримує роботу в хмарних середовищах і середовищах Інтернету речей (IoT) [11, 13, 15, 39-40]. Рішення [16, 65] спрямоване на зниження складності виявлення загроз, підвищення ефективності системи безпеки та забезпечення інтеграції з іншими продуктами, зокрема для обміну інформацією про виявлені вразливості (рис. 1.21).

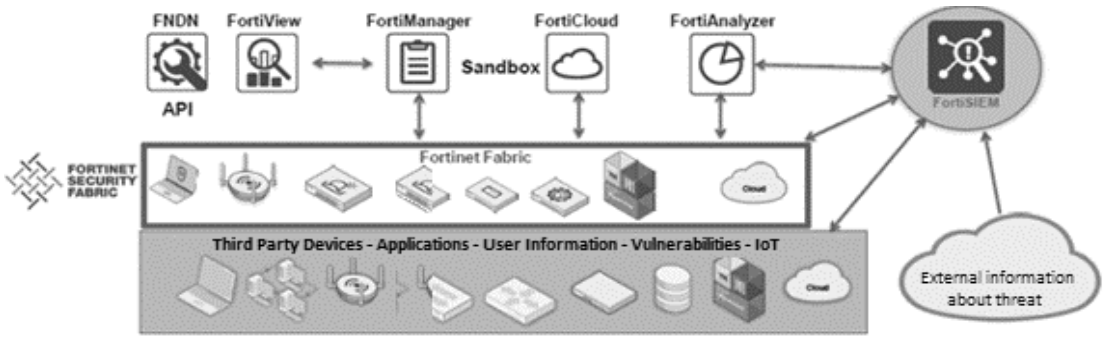


Рис. 1.21. FortiSIEM в концепції Fortinet Security Fabric

Основні переваги FortiSIEM (рис. 1.22): масштабований і гнучкий збір журналів подій; підтримка рішень управління інцидентами та системи сповіщень; наявність налаштовуваних інформаційних панелей моніторингу; інтеграція зовнішніх джерел даних про загрози; реалізація масштабованих механізмів аналізу даних; визначення базових показників і виявлення статистичних аномалій поведінки користувачів, серверів та кінцевих пристроїв; інтеграція зі сторонніми технологіями та сервісами [13].

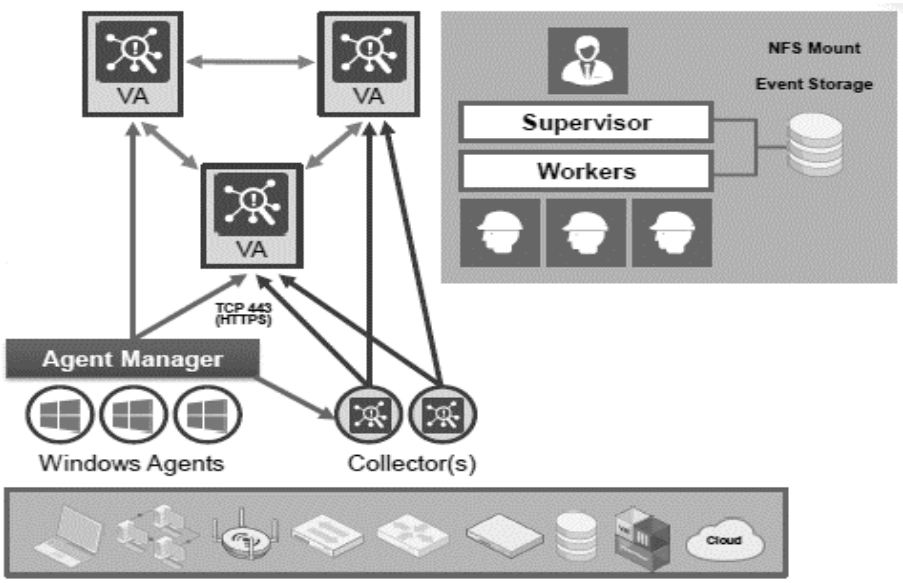


Рис. 1.22. Архітектура FortiSIEM

1.7. Ixia ThreatARMOR – це спеціалізований засіб мережевого захисту, призначений для фільтрації шкідливого трафіку на основі актуальних даних про загрози. Даний інструмент не є класичною SIEM-системою, проте може

використовуватися спільно з SIEM для підвищення ефективності виявлення та запобігання кіберзагрозам (рис. 1.23).

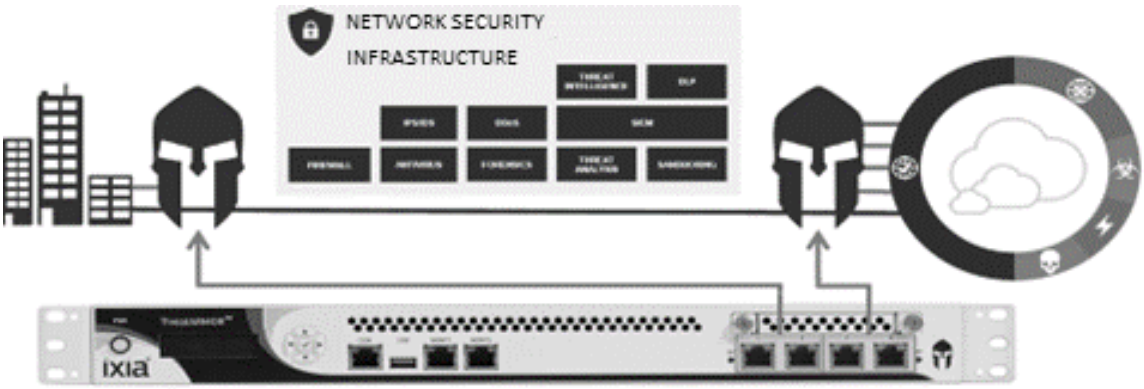


Рис. 1.23. Архітектура Ixia ThreatARMOR

Основні можливості: забезпечення високої пропускної здатності мережевого трафіку; блокування з'єднань із відомими шкідливими ресурсами та недовіреними джерелами; зменшення кількості хибнопозитивних спрацьовувань; підвищення ефективності системи безпеки за рахунок зниження навантаження на SIEM; регулярне оновлення даних про загрози з використанням хмарних сервісів; виявлення скомпрометованих внутрішніх систем; блокування підозрілих IP-адрес і з'єднань; підтримка відмовостійкості (резервування живлення, режим bypass); швидке розгортання та централізоване управління; підвищення ефективності інфраструктури мережевої безпеки.

1.8. MozDef Mozilla SIEM-система [13-14, 53, 43], з відкритим вихідним кодом, розроблена компанією Mozilla для автоматизації процесів оброблення інцидентів безпеки. Система побудована на основі мікросервісної архітектури, у якій кожен компонент функціонує як окремий сервіс у контейнеризованому середовищі (Docker), що забезпечує високу продуктивність, масштабованість і відмовостійкість (рис. 1.24).

Перевагами системи є: відсутність необхідності використання агентів, підтримка роботи зі стандартними журналами у форматі JSON; висока масштабованість завдяки мікросервісній архітектурі; підтримка інтеграції з хмарними джерелами даних, зокрема AWS CloudTrail та AWS GuardDuty.

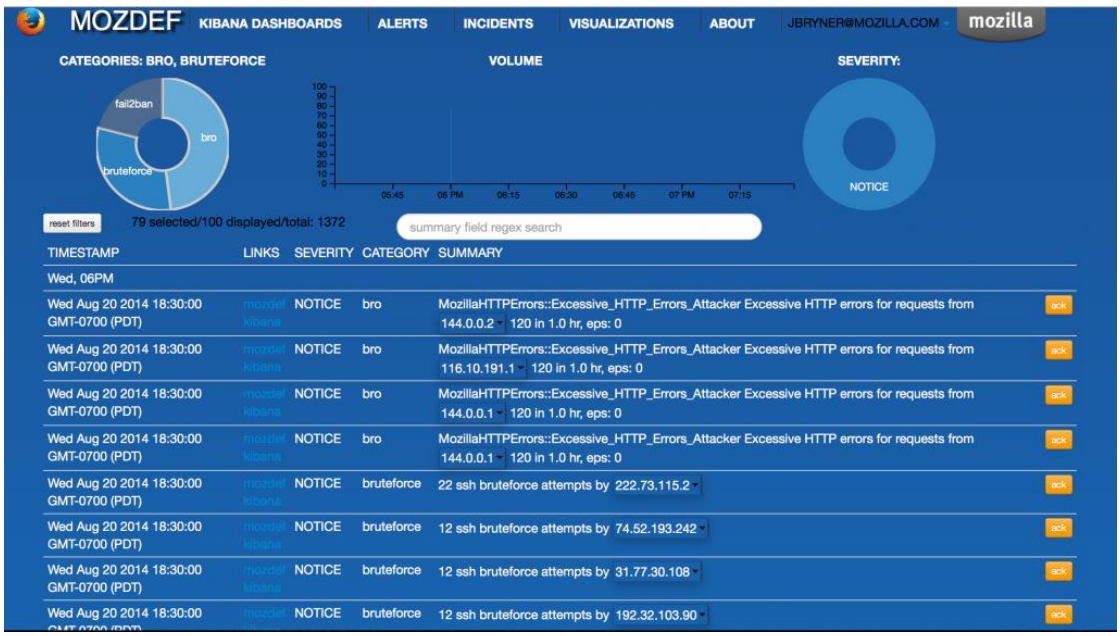


Рис. 1.24. Архітектура MozDef Mozilla

1.9. Wazuh – це система моніторингу безпеки з відкритим вихідним кодом, яка поєднує функції HIDS та SIEM і призначена для виявлення загроз, контролю цілісності та аналізу подій безпеки (рис. 1.25).

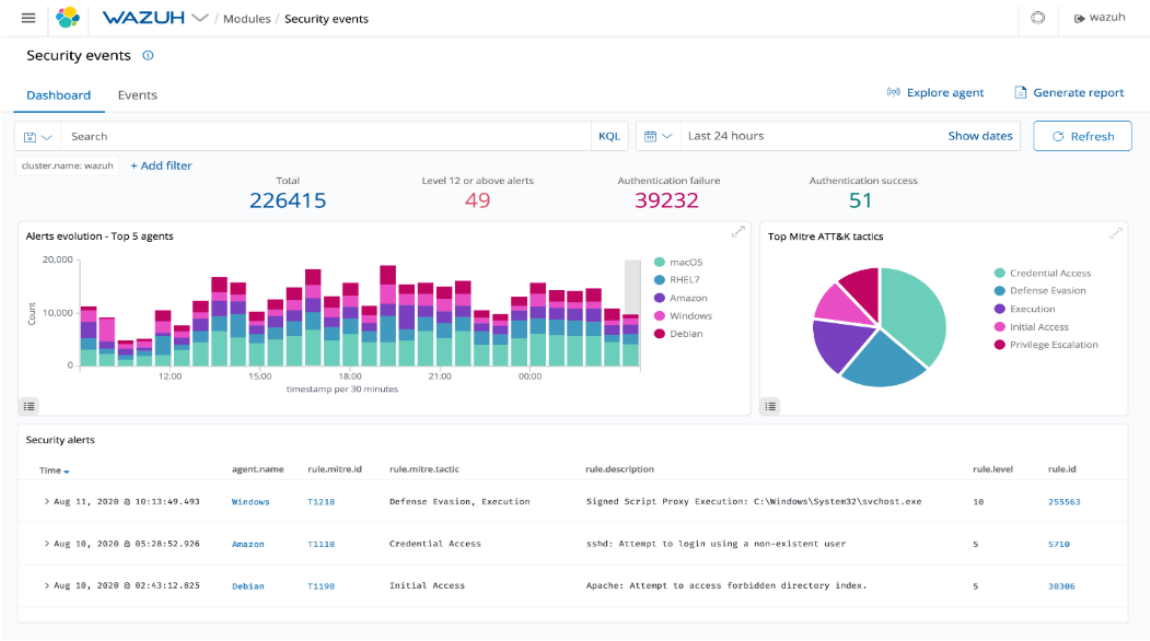


Рис. 1.25. Архітектура Wazuh

Основні переваги системи: побудована на основі проєкту OSSEC та сумісна з ним; підтримка різних способів розгортання, зокрема Docker, Puppet, Chef, Ansible; можливість моніторингу хмарних сервісів, включаючи AWS та

Azure; наявність розширеного набору правил для виявлення різних типів атак із підтримкою стандартів, зокрема PCI DSS та CIS; інтеграція із системами збору та аналізу логів, такими як Splunk, а також підтримка API для розширення функціональності [13, 32, 60].

1.10. Prelude OSS – це гнучка модульна SIEM-система з відкритим вихідним кодом, яка підтримує різні формати журналів подій та забезпечує інтеграцію зі сторонніми засобами безпеки, такими як OSSEC, Snort і система виявлення вторгнень Suricata (рис. 1.26).

Переваги системи, відповідно до [15, 32]: тривалий період розробки та використання, що свідчить про стабільність і зрілість рішення; підтримка широкого спектра форматів журналів подій; нормалізація даних до формату IDMEF (Intrusion Detection Message Exchange Format), що забезпечує ефективний обмін інформацією між різними системами безпеки.

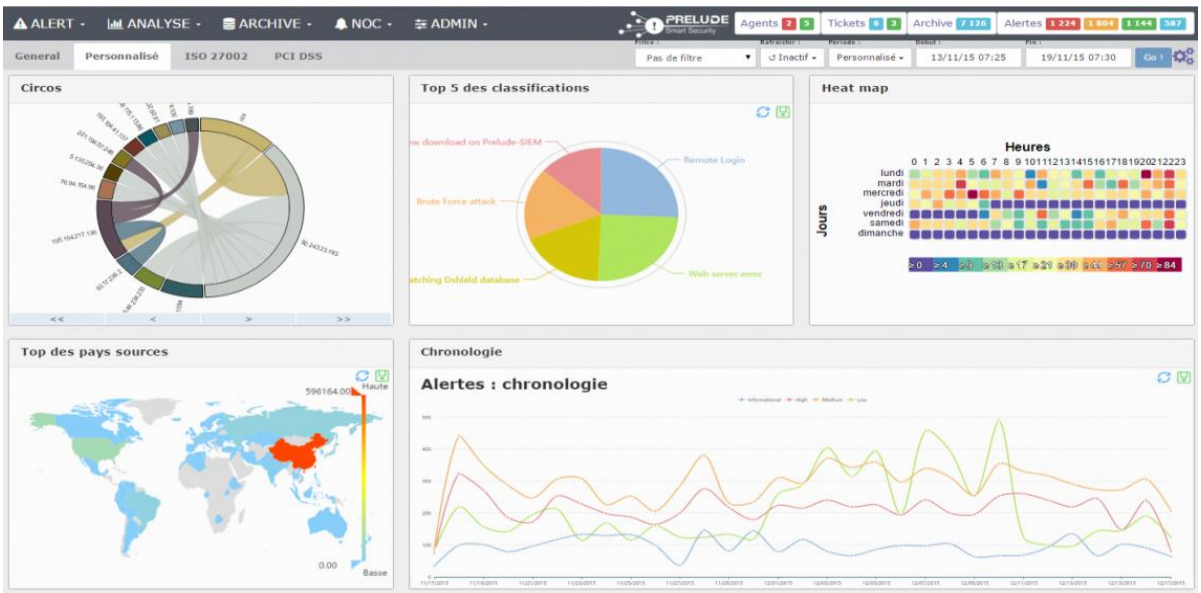


Рис. 1.26. Архітектура Prelude OSS

1.11. Sagan – це система аналізу журналів подій безпеки, яка використовується для виявлення загроз шляхом кореляції подій на основі сигнатурного підходу. Вона може інтегруватися з іншими засобами захисту інформації та використовуватися як компонент SIEM-систем (рис. 1.27).

Рис. 1.27. Архітектура Sagan

Система має такі переваги: повна сумісність із правилами та форматами бази даних Snort; підтримка сигнатурного аналізу подій безпеки; багатопотокова архітектура, що забезпечує високу продуктивність; можливість інтеграції з іншими системами моніторингу та аналізу безпеки [13, 32].

1.12. SolarWinds – це система управління подіями безпеки та журналами, яка забезпечує моніторинг, аналіз і реагування на інциденти в режимі реального часу [16, 33, 43] (рис. 1.28).

Рис. 1.28. Архітектура SolarWinds

Основні можливості системи: оперативне виявлення підозрілих дій та кіберзагроз; безперервний моніторинг стану інформаційної безпеки; аналіз і фіксація часових характеристик подій; підтримка відповідності міжнародним стандартам, зокрема PCI DSS, HIPAA, SOX, STIG, DISA; підтримка різних варіантів розгортання (локального та хмарного); можливість функціонування в середовищах Windows та Linux.

1.13. ManageEngine – це SIEM-рішення, орієнтоване на аналіз журналів подій з метою виявлення загроз, оцінювання продуктивності систем та забезпечення ІБ (рис. 1.29).

Система забезпечує моніторинг і аналіз різних компонентів інформаційної інфраструктури, зокрема веб-серверів, DHCP-серверів, систем управління базами даних, серверів друку та поштових сервісів.

Рис. 1.29. Архітектура ManageEngine

Крім того, рішення ManageEngine, яке функціонує в середовищах Windows і Linux, підтримує механізми забезпечення відповідності міжнародним стандартам захисту інформації, таким як PCI DSS, HIPAA та ISO/IEC 27001 та ін. [15, 34].

1.14. EventTracker – це SIEM-система, призначена для моніторингу подій безпеки, аналізу журналів та управління інцидентами в інформаційній інфраструктурі (рис. 1.30).

Рис. 1.30. Архітектура EventTracker

Ключові можливості системи: оповіщення в режимі реального часу та підтримка реагування на інциденти на основі правил і політик безпеки; пошук і аналіз подій із використанням методів комп'ютерної криміналістики; індексація журналів із застосуванням технології Elasticsearch, що забезпечує масштабованість і швидкий пошук; формування звітів (понад 1500 шаблонів) для аналізу безпеки та контролю відповідності стандартам; підтримка відповідності міжнародним стандартам, зокрема PCI DSS, HIPAA, ISO/IEC 27001, NIST 800-171, DoD, RMF, GDPR; аналіз поведінки користувачів і систем, а також кореляція подій у режимі реального часу; інтеграція з джерелами даних про загрози (threat intelligence) для підвищення ефективності виявлення атак [13, 35].

1.15. Trustwave SIEM Enterprise – це система управління подіями ІБ, призначена для збору, аналізу та кореляції подій безпеки з метою виявлення загроз та реагування на інциденти (рис. 1.31).

Переваги системи Trustwave [33, 43]: інтеграція з іншими рішеннями екосистеми Trustwave, що забезпечує можливість автоматизованого реагування на інциденти, зокрема ізоляцію скомпрометованих кінцевих пристроїв або

блокування облікових записів; підтримка рішень кореляції подій безпеки та аналізу загроз; відносно проста архітектура, що зменшує складність розгортання та подальшого масштабування системи; можливість централізованого моніторингу та управління подіями ІБ.

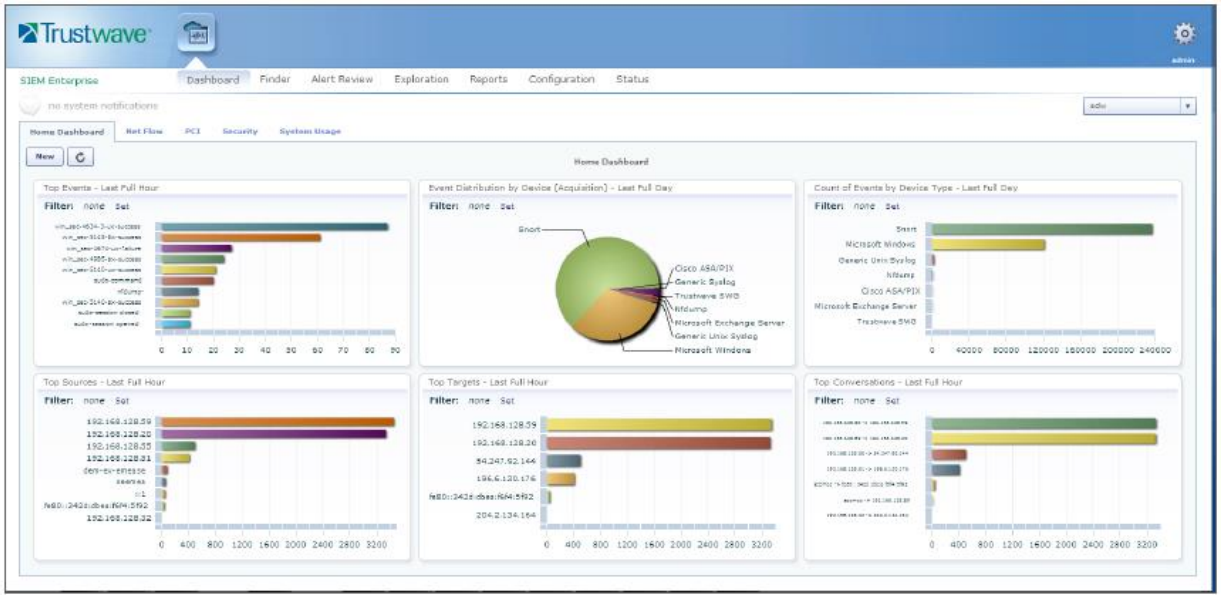


Рис. 1.31. Архітектура Trustwave SIEM Enterprise

1.16. BlackStratus SIEM Storm – це SIEM-система, призначена для моніторингу, аналізу та кореляції подій ІБ в розподілених мережах (рис. 1.32).

Система інтегрується з існуючим мережевим і безпековим обладнанням, забезпечуючи розширені можливості аналізу та реагування на загрози [13, 33, 37].

Основні можливості системи: відмовостійка архітектура з підтримкою аварійного перемикання та багаторівневого резервування; виявлення атак у режимі реального часу, включаючи потенційні атаки нульового дня, на основі кореляції подій і аналізу поведінки; інтеграція з системами виявлення вторгнень та використання даних про вразливості (зокрема CVE) для підвищення точності виявлення загроз; забезпечення централізованого моніторингу подій у розподілених мережах та виявлення аномальної активності; формування звітів для контролю відповідності стандартам інформаційної безпеки, зокрема ISO, PCI DSS, HIPAA, SOX.

Рис. 1.32. Архітектура BlackStratus SIEM Storm

У табл. 1.4, відповідно до [13, 37, 41, 43, 53, 60-61], систематизовано та представлено детальний аналіз SIEM-систем за такими 18 критеріями (запропоновані автором): 1. Аудит та перевірка на відповідність стандартам; 2. Повноцінність системи (повноцінна – «+», є лише обробка логів – «-»); 3. Оцінка захищеності ресурсів системи, що контролюється (у т.ч. КВР); 4. Перевірка відповідності системи управління ІБ; 5. Управління ризиками ІБ; 6. Збір та зберігання подій, які надходять до системи; 7. Обробка та аналіз зареєстрованих подій; 8. Виявлення атак та порушень політик безпеки; 9. Виявлення та розбір інцидентів безпеки; 10. Можливість розслідувань; 11. Пошук уразливостей; 12. Формування звітів; 13. Підтримка роботи з хмарними середовищами; 14. Підтримка роботи з Big Data платформами; 15. Можливість інтеграції з новими системами у майбутньому; 16. Розширені можливості пошуку; 17. User Friendly інтерфейс; 18. Можливість безкоштовного використання.

Таблиця 1.4

Порівняльний аналіз SIEM-систем

№	Назва	Критерії																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1.	IBM QRadar	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	-
2.	LogRhythm	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	-
3.	Splunk	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	-
4.	McAfee (ESM)	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	-
5.	AlienVault USM	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	-
6.	FortiSIEM	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	-
7.	Ixia ThreatARMOR	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	+	+	-
8.	MozDef	+	+	-	+	-	+	+	+	+	+	-	+	+	+	+	+	+	+
9.	Wazuh	+	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+
10.	Prelude OSS	+	+	-	+	+	+	+	+	+	+	+	+	-	-	-	+	+	+
11.	Sagan	-	-	-	-	-	+	+	+	+	+	-	+	-	-	-	+	-	+
12.	SolarWinds	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	-
13.	ManageEngine	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	-
14.	EventTracker	+	+	-	+	-	+	+	+	+	+	+	+	+	+	-	+	+	-
15.	Trustwave SIEM Enterprise	+	+	-	+	+	+	+	-	+	+	+	+	+	+	-	+	+	-
16.	Black Stratus SIEM Storm	+	+	-	+	-	+	+	+	+	+	+	+	-	-	-	+	+	-

У табл. 1.4 наведено результати порівняльного аналізу сучасних SIEM-систем та суміжних рішень у сфері ІБ. Результати аналізу показали, що найбільш функціонально повними є системи IBM QRadar, LogRhythm, Splunk, McAfee (ESM), AlienVault USM, FortiSIEM, SolarWinds та ManageEngine, які відповідають більшості визначених критеріїв.

Зазначені рішення характеризуються високим рівнем інтеграції, розвиненими механізмами аналізу подій та підтримкою сучасних стандартів ІБ, проте відрізняються за вартістю впровадження та експлуатації.

Водночас проведений аналіз засвідчив, що жодна з розглянутих систем не забезпечує повного спектру всіх визначених критеріїв. З урахуванням зазначеного, вбачається за доцільне розробити універсальну

SIEM-систему, у якій будуть враховані всі перелічені функціональні особливості та переваги.

1.6. Формалізація завдання дисертаційного дослідження

Таким чином, у першому розділі, на основі проведеного аналізу (рис. 1.33 Етап 1), визначено і обґрунтовано основні задачі дослідження (рис. 1.33 Етап 2-9), розв'язання яких необхідне для досягнення мети, що поставлена в дисертаційній роботі.

Таким чином у наступних розділах роботи згідно поставленої мети буде розроблено: структурно-аналітичну модель оброблення даних для інтелектуалізованого виявлення аномалій у хмарних системах ІКС (п. 2.1); модель онтологіко-реляційного сховища даних для зберігання та оброблення великих обсягів інформації (п. 2.2); модель інтеграційної шини даних для розподілу навантаження та безперервного обміну даними (п. 3.1); систему корелювання подій та управління ІТ-інцидентами для формалізації інформаційної технології, що реалізує процеси управління ІТ-інцидентами на ОКІ (п. 3.2).

Рис. 1.33. Етапи виконання наукового дослідження

Крім того, на основі дата сету NSL-KDD навчено нейронну мережу з точки зору виявлення аномалій типу DoS, U2R, R2L та Probe (п. 4.1); створено методику зберігання та класифікації даних, яка дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірностей та забезпечувати високу швидкість пошуку (п. 4.2); сформовано специфікацію реалізації SIEM-систем на ОКІ, у вигляді основних та додаткових вимог (п. 4.3); розроблено спеціальний програмний застосунок, який можна використовувати для управління IT-інцидентами, які виникають в КІ і мають вплив на КВР (п. 4.4).

1.7. Висновки до першого розділу

Проведено аналіз сучасних підходів до управління IT-інцидентами на ОКІ для виявлення їх переваг та недоліків. За результатами проведеного аналізу підходів до виявлення аномалій хмарному середовищі встановлено, що загальна мета всіх досліджень полягає в тому, щоб швидко ідентифікувати аномалії та не знизити рівень неправильної класифікації. Крім того, кожен метод виявлення має свої переваги та працює краще для певних наборів даних, але жоден не є універсальним і не може виявити всі сто відсотків шкідливих програм. Аналіз сучасних типів баз даних, що використовуються в SIEM-системах, показав, що кожен з їх видів залишається актуальним у власній сфері, де взаємозв'язки між даними обумовлені конкретною структурою СУБД, а використання однієї бази даних для всіх цих задач не відповідає вимогам архітектури та безпеки. Крім того, проаналізовані існуючі на ринку рішення інтеграційних шин даних, та встановлено, що кожні з них мають свої особливості та суттєві відмінності, які визначають їх сферу використання, а також відрізняються функціоналом, додатковими налаштуваннями та вартістю ліцензії. Також систематизовано та представлено детальний аналіз 16 SIEM-систем за 18 запропонованими критеріями. Зокрема відображено їх функціональність, основний принцип роботи, а також проведено порівняльний аналіз їх можливостей та відмінностей, переваг та недоліків використання, та відповідності до міжнародних

специфікацій та стандартів. Проведений аналіз дозволив формалізувати завдання дисертаційного дослідження щодо розробки і вдосконалення системи корелювання подій та управління ІТ-інцидентами на ОКІ.

1.8. Список літератури до першого розділу

1. Закон України «Про критичну інфраструктуру», від 16.11.2021 № 1882-IX.

2. Закон України «Про основні засади забезпечення кібербезпеки України», від 05.10.2017 № 2163-VIII.

3. Постанова Кабінету Міністрів України «Про затвердження Загальних вимог до кіберзахисту об’єктів критичної інфраструктури», від 19 червня 2019 р. № 518.

4. Постанова Кабінету Міністрів України «Деякі питання об'єктів критичної інфраструктури», від 9 жовтня 2020 р. № 1109.

5. Команда реагування на комп’ютерні надзвичайні події України: [Електронний ресурс]. – Режим доступу: https://cert.gov.ua.

6. Behl D., Behl K. Cybersecurity and Cyberwar: What Everyone Needs to Know. – Oxford: Oxford University Press, 2017.

7. Bogachuk I., Sokolov V., Buriachok V. «Monitoring subsystem for wireless systems based on miniature spectrum analyzers», Proceedings of the International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2018, pp. 581–585. DOI: 10.1109/infocommst.2018.8632151.

8. Vladymyrenko M., Sokolov V., Buriachok V., Platonenko A., Ageyev D. «Analysis of Implementation Results of the Distributed Access Control System», 2019 IEEE International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), Kyiv, Ukraine, 2019, pp. 1–6. DOI: 10.1109/PICST47496.2019.9061376.

9. Buriachok V., Sokolov V., Skladannyi P. «Security rating metrics for distributed wireless systems», IEEE International Scientific-Practical Conference

on Problems of Infocommunications Science and Technology (PIC S&T), 2019, pp. 612–616. DOI: 10.1109/PICST47496.2019.9061357.

10. Aslan Ö., Ozkan-Okay M., Gupta D. «Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment», IEEE Access, vol. 9, 2021, pp. 83252–83271.

11. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. «A concept of the architecture and creation for SIEM system in critical infrastructure». Studies in Systems, Decision and Control. 2021. Vol. 346. P. 221–242.

12. Berdibayev R., Gnatyuk S., Tynymbayev S., Sydorenko V. «Advanced technologies of cyber incident management in critical infrastructure». Kyiv: Pro Format, 2022, 125 p.

13. Gnatyuk S., Berdibayev R., Fesenko A., Kyryliuk O., Bessalov A. «Modern SIEM Analysis and Critical Requirements Definition in the Context of Information Warfare», CEUR Workshop Proceedings, 2021, vol. 3188, pp. 149–166.

14. Karlzén H. «An Analysis of Security Information and Event Management Systems». Göteborg: Chalmers University of Technology, 2009. – Режим доступу: http://publications.lib.chalmers.se/records/fulltext/89572.pdf

15. Lee J., Kim Y., Kim J., Kim I. «Toward the SIEM architecture for cloud-based security services», 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, 2017, pp. 398–399. DOI: 10.1109/CNS.2017.8228696.

16. Vielberth M., Pernul G. «A security information and event management pattern», Proceedings of the 12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP), 2018, pp. 1–12.

17. Vielberth M., Pernul G. «Security information and event management: analysis, trends and usage in critical infrastructures», Computers & Security, 2018, vol. 77, pp. 1–14. DOI: 10.1016/j.cose.2018.03.001.

18. Mid-Year Update: 2022 SonicWall Cyber Threat Report, 39 p.

19. Xiao F., Lin Z., Sun Y., Ma Y. «Malware detection based on deep learning of behavior graphs», Mathematical Problems in Engineering, 2019.

20. Бобровнікова К.Ю., Денисюк Д.О. «Метод виявлення шкідливого програмного забезпечення шляхом аналізу мережного трафіку та поведінки програмного забезпечення в комп'ютерних системах», Вісник Хмельницького національного університету, 2020, т. 1, № 4 (287), с. 7–11.

21. Mirza Q.K.A., Awan I., Younas M. «A cloud-based energy efficient hosting model for malware detection framework», Proceedings of the IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6.

22. Zhou W., Yu B. «A cloud-assisted malware detection and suppression framework for wireless multimedia system in IoT based on dynamic differential game», China Communications, 2018, vol. 15, no. 2, pp. 209–223.

23. Indirapriyadarsini P., Mohiuddin M.U., Taqueeuddin M., Reddy C.S., Koushik T. «Malware detection using machine learning and cloud computing», International Journal of Research in Applied Sciences and Engineering Technology, 2020, vol. 8, no. 6, pp. 101–104.

24. Deyannis D., Papadogiannaki E., Kalivianakis G., Vasiliadis G., Ioannidis S. «TrustAV: Practical and privacy preserving malware analysis in the cloud», Proceedings of the 10th ACM Conference on Data and Application Security and Privacy, 2020, pp. 39–48.

25. Gnatyuk S., Satybaldiyeva F., Sydorenko V., Zhyharevych O., Polozhentsev A. «Model of Information Technology for Efficient Data Processing in Cloud-based Malware Detection Systems of Critical Information Infrastructure», CEUR Workshop Proceedings, 2023, vol. 3421, pp. 206–213.

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

27. Melnyk V., Bahnyuk N., Melnyk K., Zhyharevych O., Panasyuk N. «Implementation of the simplified communication mechanism in the cloud of high performance computations», East-European Journal of Enterprise Technologies, 2017, no. 2/2(86), pp. 24–32.

28. Melnyk V., Pekh P., Melnyk K., Bahnyuk N., Zhyharevych O. «Design and implementation of interdomain communication mechanism for high performance

data processing», East-European Journal of Enterprise Technologies, 2016, no. 1(9), pp. 10–15.

29. Agrawal K., Makwana H. «A study on critical capabilities for security information and event management», International Journal of Science and Research (IJSR), 2015, vol. 4, no. 7, pp. 1893–1896.

30. Ribolovlev D., Karasov S., Polyakov S. «Classification of emergency management systems for incidents without backing», Food of Cyber Security, 2018, no. 3(27), pp. 47–53.

31. Ariel Query Language Guide. IBM QRadar 7.3.3 (2013, 2019). – Режим доступу:https://www.ibm.com/docs/en/SS42VS_7.3.3/com.ibm.qradar.doc/b_qradar_aql.pdf

32. SIEM Analytics. – Режим доступу: http://www.siem.su/compare_SIEM_systems.php

33. Bachane I., Adsi Y.I.K., Adsi H.C. «Real time monitoring of security events for forensic purposes in cloud environments using SIEM», 2016 Third International Conference on Systems of Collaboration (SysCo), 2016, pp. 1–3. DOI: 10.1109/SYSCO.2016.7831327.

34. AlSabbagh B., Kowalski S. «A framework and prototype for a socio-technical security information and event management system (ST-SIEM)», 2016 European Intelligence and Security Informatics Conference (EISIC), 2016, pp. 192–195. DOI: 10.1109/EISIC.2016.049.

35. Serckumecka A., Medeiros I., Bessani A. «Low-cost serverless SIEM in the cloud», 2019 38th Symposium on Reliable Distributed Systems (SRDS), 2019, pp. 381–391. DOI: 10.1109/SRDS47363.2019.00057.

36. Nabil M., Soukainat S., Lakbabi A., Ghizlane O. «SIEM selection criteria for an efficient contextual security», 2017 International Symposium on Networks, Computers and Communications (ISNCC), 2017, pp. 1–6. DOI: 10.1109/ISNCC.2017.8072035.

37. Mahmoud R.-V., Kidmose E., Turkmen A., Pilawka O., Pedersen J.M. «DefAtt – architecture of virtual cyber labs for research and education», 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2021, pp. 1–7.

38. Danik Yu., Hryschuk R., Gnatyuk S. «Synergistic effects of information and cybernetic interaction in civil aviation», Aviation, 2016, vol. 20, no. 3, pp. 137–144.

39. Oksiiuk O., Chaikovska V., Fesenko A. «Security technique for authentication process in the cloud environment», 2019 IEEE International Scientific-Practical Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 379–382. DOI: 10.1109/PICST47496.2019.9061248.

40. Gnatyuk S., Berdibayev R., Avkurova Z., Verkhovets O., Bauyrzhan M. «Studies on cloud-based cyber incidents detection and identification in critical infrastructure», CEUR Workshop Proceedings, 2021, vol. 2923, pp. 68–80.

41. Lukova-Chuiko N., Fesenko A., Papirna H., Gnatyuk S. «Threat hunting as a method of protection against cyber threats», CEUR Workshop Proceedings, 2021, vol. 2833, pp. 103–113.

42. Astapenya V., Buriachok V., Sokolov V., Skladannyi P., Ageyev D. «Last mile technique for wireless delivery system using an accelerating lens», Proceedings of the IEEE International Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2020, pp. 811–814. DOI: 10.1109/PICST51311.2020.946788.

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

44. Жигаревич О.К., Бердибаєв Р.Ш., Сидоренко В.М., Положенцев А.А., Кримська А.О. «Модель онтологіко-реляційного сховища даних для функціонування хмарної SIEM-системи», Проблеми інформатизації та управління. 2023. Т. 4. № 76. С. 17-27.

45. Гнатюк С.О., Бердибаєв Р.Ш., Богун А.М., Сидоренко В.М., Положенцев А.А., Жигаревич О.К. «Інтеграційна шина даних для ефективного функціонування системи управління подіями інформаційної безпеки», Проблеми інформатизації та управління. 2023. Т. 3. № 75. С. 29-40.

46. Gnatyuk, S., Berdibayev, R., Sydorenko, V., Polozhentsev, A., Ryabyy, M. «Enterprise Service Bus Construction in SOA Architecture for SIEM Implementation in Critical Information Infrastructure» CEUR Workshop Proceedings, 2022, Vol. 3288, Paper 2, P. 11–20.

47. Kipchuk F. et al. «Investigation of availability of wireless access points based on embedded systems», 2019 IEEE International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2019. DOI: 10.1109/PICST47496.2019.9061551.

48. Жигаревич О.К., Медведєв М.В. «Інформаційна ситема “Студент - ФКНІТ” засобами PHP». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2017. № 26. С. 88–92.

49. Жигаревич О.К., Котлярець В.В., Луць А.Р. «Модель екосистеми навчального програмного забезпечення». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2017. № 26. 167-177.

50. Жигаревич О.К., Мельник В.М., Мельник К.В. «Підтримка оголошеної/встановленої комунікації в мережі через стандартні сокети API». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2015. № 19. С. 23–27.

51. Мельник К.В., Мельник В.М., Багнюк Н.В., Жигаревич О.К., Климюк М. «Система попереднього відбору кандидатів на основі нечіткої логіки». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2015. № 19. С. 114–120.

52. Дмитрук Я.В., Гришанович Т.О., Глинчук Л.Я., Жигаревич О.К. «Кібервійна як різновид інформаційних війн. Захист кіберпростору України». Кібербезпека: освіта, наука, техніка. 2022. Т. 4. №16. С. 28-36.

53. Miller D., Harris S., Harper A., VanDyke S., Blask C. «Security information and event management (SIEM) implementation». McGraw-Hill Osborne Media, 2010.

54. Sreemathy J., Joseph V.I., Nisha S., Prabha I.C., Priya R.M.G. «Data integration in ETL using Talend», 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1444–1448.

55. Mkhwanazi X., Le H., Blake E. «Clustering between data mules for better message delivery», 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 209–214.

56. Kumara I., Gamage C. «Towards reusing ESB services in different ESB architectures», 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, 2010, pp. 25–30.

57. Pobochenko L., Prokopieva A., Zhyharevych O., Gavrylko O., Panikar G., Gavrilko T. «Risks of investing in FinTech at the global and national levels». CEUR Workshop Proceedings. Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2025), June 20 - 22, 2025, Kyiv, Ukraine, 2025. Vol. 4024. P. 468-478.

58. Zdolbitska N., Ostapchuk O., Lavrenchuk S., Terletskyi T., Kaidyk O., Zhyharevych O. «Business information system for forecasting raw material stocks for the production of flexible packaging». 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2024, P. 1-8.

59. Sydorenko V., Zhyharevych O., Berdybaev R., Polozhentsev A., Fesenko A. «Ontological-Relational Data Store Model for a Cloud-based SIEM System Development». CEUR Workshop Proceedings. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2024), February 28, 2024, Kyiv, Ukraine, 2024. Vol. 3654. P. 343-354.

60. Gnatyuk S., Berdibayev R., Aleksander M., Sydorenko V., Zhyharevych O., Polozhentsev A. «Software System for Cybersecurity Events Correlation and Incident Management in Critical Infrastructure». Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies. 2024. Vol. 213. P. 247-269. Springer, Cham.

61. Polozhentsev A., Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O. «Novel Cyber Incident Management System for 5G-based Critical Infrastructures». IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, P. 1037-1041.

62. Gnatyuk S., Berdibayev R., Azarov I., Baisholan N., Lozova I. «Modern Types of Databases for SIEM System Development». CEUR Workshop Proceedings, 2021, vol. 3187, pp. 127-138.

63. Сидоренко В., Положенцев А., Юдін О., Жигаревич О. «Функціональна модель визначення критичності галузевих інформаційно-телекомунікаційних систем». АВІА-2023: XVI міжнар. наук.-техніч. конф., 18-20 квітня 2023 р.: тези доп., Київ: НАУ, 2023. С. 16.14-16.17.

64. Gnatyuk S., Sydorenko V., Yudin O., Zhyharevych O., Polozhentsev A. «Method for Calculating the Criticality Level of Sectoral Information and Telecommunication Systems». CEUR Workshop Proceedings, Proceedings of the: Information Technology and Implementation (IT&I2022), November 30 - December 02, 2022, Kyiv, Ukraine, Vol. 3347, Paper 20, P.234-245.

65. Gnatyuk S., Zhaksigulova D., Zhyharevych O., Ospanova D., Chuba I. «Studies on WSN Models for IoT-based Monitoring Systems in the Critical Infrastructure of the State». CEUR Workshop Proceedings, Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2023-II), October 26, 2023, Kyiv, 2023, Vol. 3550, P. 167-180.

РОЗДІЛ 2
РОЗРОБКА МОДЕЛЕЙ ДЛЯ УПРАВЛІННЯ ІТ-ІНЦИДЕНТАМИ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

2.1. Структурно-аналітична модель оброблення даних в критичній інформаційній інфраструктурі

Проведені у першому розділі та в роботах [1-13] дослідження процесу збирання, зберігання та обробки метаданих у хмарних системах виявлення аномалій показали, що загальну структуру моделі обробки даних можна представити у вигляді схеми рис. 2.1.

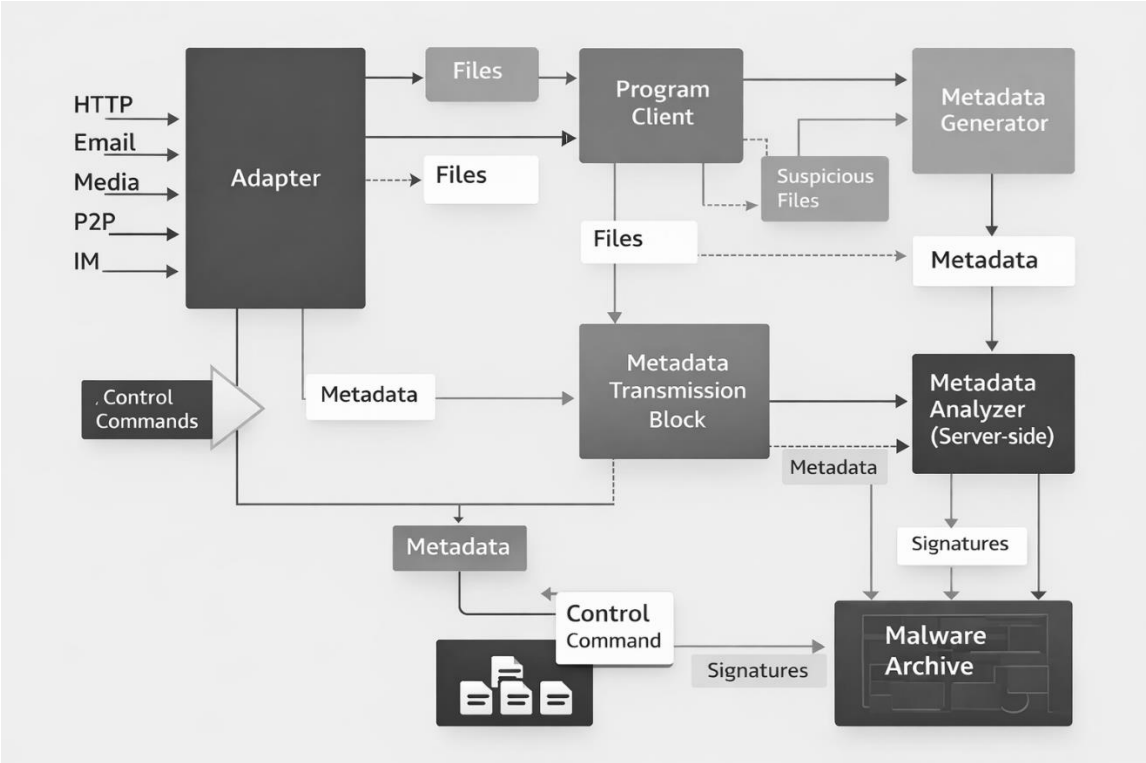


Рис. 2.1. Структурна схема обробки даних у хмарних системах виявлення аномалій

Розглянемо детальніше призначення основних функціональних блоків моделі оброблення даних [1].

Потік даних із каналів зв'язку надходить на ***телекомунікаційний адаптер*** (мережевий додаток), основним завданням якого є виділення окремих додатків із загального потоку даних, формування файлів (команд

керування передачею) для подальшої обробки програмним клієнтом, а також забезпечення безперешкодної передачі метаданих у канали зв’язку телекомунікаційної мережі.

Програмний клієнт є модулем, розміщеним на стороні користувача, який забезпечує взаємодію апаратної та програмної складових системи. Його функції включають передачу підозрілих файлів до формувача метаданих, а також представлення результатів роботи хмарної системи виявлення аномалій у зручному вигляді (таблиці, графіки, діаграми тощо). Програмний клієнт функціонально пов’язаний з аналізатором файлів – програмним комплексом, призначеним для виконання попереднього сигнатурного та евристичного аналізу (порівняння зі зразками, перевірка допустимості значень тощо).

Формувач метаданих призначений для виділення спеціальних сигнатур у підозрілих файлах із використанням сучасних методів хешування. Сформовані сигнатури через телекомунікаційний адаптер передаються в канал зв’язку мережі. Передача метаданих через проміжні вузли комутації (блок передачі метаданих) здійснюється відповідно до стандартних протоколів і сучасних методів управління інформаційним трафіком.

Аналізатор метаданих у хмарній системі виявлення аномалій виконує функції виявлення загроз, перевірки достовірності прийнятих рішень та ідентифікації джерел поширення загроз. Виявлені джерела проходять додаткову автоматичну перевірку для мінімізації хибних спрацьовувань. Отримана інформація про нові загрози та джерела їх поширення заноситься до архіву шкідливого програмного забезпечення та стає доступною іншим користувачам системи.

Інформація про інциденти використовується для самонавчання аналізатора метаданих, що забезпечує оперативну реакцію на нові типи атак та автоматичне виявлення активних загроз. При цьому враховуються результати сигнатурного та евристичного аналізу.

Завдяки збору та обробленню даних про підозрілу активність від усіх учасників мережі, хмарна система функціонує як розподілена експертна система, орієнтована на виявлення та аналіз кіберзагроз. Інформація, необхідна для блокування атак, оперативно поширюється між усіма вузлами мережі, що дозволяє запобігати подальшому розповсюдженню загроз.

Проведені дослідження показали, що для реалізації багатокористувацьких розподілених додатків (зокрема, хмарних систем виявлення аномалій) необхідно використовувати інтерфейс сокетів.

Сокети є програмними кінцевими точками мережевого з'єднання, що забезпечують обмін даними між комп'ютерами. Їх функціонування базується на протоколах стеку TCP/IP із використанням транспортних портів операційних систем. Сокети поділяються на три основні типи [14].

Клієнтські сокети ініціюють встановлення з'єднання, використовуючи IP-адресу віддаленого вузла та номер порту серверного сокета. Серверні сокети забезпечують встановлення з'єднання у відповідь на запит клієнта, тоді як слухаючі сокети обробляють запити на підключення, формуючи чергу з'єднань. Після звільнення серверного сокета від попередніх операцій відбувається обробка нового запиту та встановлення з'єднання [15-16].

На основі викладеного представимо математичну формалізацію структурно-аналітичної моделі оброблення даних у хмарних системах виявлення аномалій та визначимо основні часові характеристики відповідних процесів Відповідно до [1].

Етап 1. Представлення основних часових характеристик метаданих для незалежних випадкових величин.

Представимо математичну формалізацію моделі та визначимо основні часові характеристики цих процесів. Час оброблення метаданих в аналізаторі хмарних систем виявлення аномалій (програмним сервером) представимо у вигляді суми незалежних випадкових величин $\xi_1, \xi_2, ..., \xi_N$, які мають однаковий розподіл F з виробляючою функцією моментів $M(s)$.

Етап 2. Знаходження розподілу виробляючої функції моментів $\chi(s)$.

Нехай N – цілочисленна випадкова величина з виробляючою функцією $A(s)=\sum P_i s^i$, яка є незалежною від усіх ξ_j. Тоді випадкова сума $S=\xi_1+...+\xi_N$ має розподіл, що описується виробляючою функцією моментів:

$$\chi(s)=W(M(s)), \tag{2.1}$$

де $W(s)$ – виробляюча функція, що описує випадкове число запитуваних програмним клієнтом елементів метаданих, $M(s)$– виробляюча функція моментів, що характеризує випадковий час обробки одного елемента метаданих.

Етап 3. Знаходження виробляючої функції моментів $M(s)$.

Кількість елементів метаданих, запитуваних програмним клієнтом, опишемо рівномірним дискретним розподілом із цілими значеннями в межах від h до ℓ. За умови рівноймовірності подій з імовірністю $\overline{p}$, виробляюча функція моментів має вигляд:

$$M(s)=\overline{p}(e^{hs}+e^{(h+1)s}+...+e^{(l-1)s}+e^{ls})=\frac{(\overline{p}(e^{hs}-e^{(l+1)s}))}{(1-e^{s})}.$$

Етап 4. Знаходження виробляючої функції, запитуваних елементів метаданих $W(s)$.

Виробляюча функція цього розподілу:

$$W(s)=\frac{(\overline{p}(s^{h}-s^{(\ell+1)}))}{(1-s)}.$$

Для оцінювання випадкового часу обробки одного елемента метаданих використовуємо рівномірний безперервний розподіл з параметрами a і b. Тоді відповідно до (2.1) виробляюча функція моментів сумарного часу оброблення $\chi(s)$ визначається як:

$$\chi(s) = \overline{p}\left(\frac{\left(\frac{e^{as}-e^{bs}}{(a-b)s}\right)^{h} - \left(\frac{e^{as}-e^{bs}}{(a-b)s}\right)^{\ell+1}}{1-\frac{e^{as}-e^{bs}}{(a-b)s}}\right). \quad (2.2)$$

Етап 5. Визначення першого μ_1 і другого μ_2 моменту часу виконання керуючої команди.

Диференціюючи $\chi(s)$ по змінній s і прирівнюючи в отриманих виразах величину s нулю, отримуємо перший μ_1 і другий μ_2 моменти щодо початку координат і, відповідно, середнє значення t_s та дисперсію D часу обробки одного елемента метаданих, переданих на запит програмного клієнта.

$$\mu_1 = t_{cp}^{(o)} = \frac{\partial(\chi(s))}{\partial s}\bigg|_{s=0} = \frac{(h+\ell)(a+b)}{4}, \quad (2.3)$$

$$J^{(o)} = \mu_2 - \mu_1^2 = \frac{\partial(\chi(s))}{\partial s^2}\bigg|_{s=0} - \left(\frac{\partial(\chi(s))}{\partial s}\bigg|_{s=0}\right)^2 = \frac{(h+\ell)(b-a)^2}{24}. \quad (2.4)$$

У випадку, коли аналізатор метаданих виконує обробку файлів різних незалежних інформаційних потоків, кількість вимог програмного клієнта на формування, аналіз та оброблення керуючих команд, доцільно описати бути розподілом Пуассона [17-19]. В такому випадку функція розподілу має вигляд $W(s) = e^{\lambda s - \lambda}$. Відповідно, виконуюча функція моментів часу формування команд та виконання команд визначається як:

$$\chi(s) = e^{\left(-\lambda+\lambda\frac{e^{as}-e^{bs}}{(a-b)s}\right)}. \quad (2.5)$$

Етап 6. Розрахунок середнього часу виконання керуючої команди t_s та її дисперсії D для обробки одного елемента метаданих, переданих на запит програмного клієнта.

З виразу (2.5) знаходимо середній час виконання завдання формування керуючої команди та його дисперсію:

$$t_{cp}^{(\phi)} = \frac{\lambda(a+b)}{2}, \tag{2.6}$$

$$J^{(\phi)} = \frac{\lambda(a^2+ab+b^2)}{3}. \tag{2.7}$$

Таким чином, розроблено структурно-аналітичну модель обробки даних, що дозволяє оцінити тимчасові характеристики обробки одного елемента метаданих та вироблення керуючої команди. Її відмінною особливістю є врахування необхідності формування команд передачі управління програмному клієнту ІКС, що загалом підвищить точність результатів математичного моделювання в умовах, що розглядаються.

2.2. Модель онтологіко-реляційного сховища даних

Після проведеного у першому розділі, а також у роботах [20–34], дослідження встановлено, що кожен тип СУБД залишається актуальним у відповідній предметній області, де взаємозв’язки між даними визначаються особливостями їх структурної організації.

Водночас доцільно розглянути можливість використання гібридних підходів до організації сховищ даних, зокрема поєднання реляційних (SQL) та нереляційних (NoSQL) СУБД. Такий підхід дозволяє забезпечити зручність структурованого зберігання та класифікації даних, а також досягти високої швидкості оброблення значних обсягів інформації за рахунок ефективних механізмів індексації та розподілу даних.

Крім того, результати аналізу [21, 32] свідчать про необхідність розроблення нової моделі сховища даних, яка повинна забезпечувати вимоги сучасних ІКС. З одного боку, така модель має забезпечувати високошвидкісне збирання, зберігання, оброблення та пошук подій безпеки у журналах. З іншого боку, вона повинна гарантувати надійне, структуроване

та узгоджене зберігання службової інформації, зокрема даних користувачів, метаданих, параметрів конфігурації, лічильників гешованих потоків, а також архіву попереджень та інцидентів.

Під поняттям онтологія будемо розуміти, формалізовану модель предметної області, що визначає множину сутностей (подій, інцидентів, ресурсів, користувачів), їх властивостей і взаємозв'язків та забезпечує семантичну інтеграцію й узгоджене оброблення даних у системах корелювання подій і управління IT-інцидентами [21].

Розробка моделі онтологіко-реляційного сховища даних.

Для обґрунтування вибору найбільш ефективних СУБД, які використовуються в сучасних SIEM-системах, запропоновано процедуру, що включає наступні етапи:

Етап 1. Введення. множини баз даних.

Етап 2. Введення множини чинникових ознак (критеріїв).

Етап 3. Процедура ранжування баз даних.

Крок 3.1. Побудова матриць попарних порівнянь.

Крок 3.2. Побудова загального вектору критеріїв та оцінка ваги векторів відносно важливості кожного критерію.

Крок 3.3. Аналіз узгодженості матриці попарних порівнянь.

Крок 3.4. Побудова матриць вектору критеріїв та оцінка ваги векторів для кожного елемента множини баз даних DB та множини елементів критеріїв EC.

Крок 3.5. Визначення рангу найбільш ефективних баз даних DB.

Етап 4. Введення множини облікових задач.

Етап 5. Результат вибору найбільш ефективних баз даних.

Розглянемо запропонований підхід більш детально.

Етап 1. Введення множини баз даних.

Введемо множину баз даних **DB** у наступному вигляді:

$$\mathbf{DB} = \{\bigcup_{i=1}^{n} DB_i\} = \{DB_1, DB_2, \ldots, DB_n\}, \tag{2.8}$$

де $DB_i \subseteq \mathbf{DB}\ (i = \overline{1,n})$ – різновиди СУБД, які використовуються в певних SIEM-системах, n – загальна кількість баз даних.

Етап 2. Введемо множину чинникових ознак (критеріїв).

Для оцінювання ефективності баз даних **DB** визначимо множину чинникових ознак (критеріїв). Запропоновані критерії **EC** представимо у наступному вигляді:

$$\mathbf{EC} = \{\bigcup_{j=1}^{q} EC_j\} = \{EC_1, EC_2, \ldots, EC_q\}, \tag{2.9}$$

де $EC_j \subseteq \mathbf{EC}\ (j = \overline{1,q})$ – категорія критеріїв для оцінювання найефективніших СУБД, q – загальна кількість критеріїв.

Етап 3. Процедура ранжування баз даних.

Крок 3.1. Побудова матриць попарних порівнянь.

Введемо матрицю A розміром $n \times n$, де кожен елемент a_{ij} представляє відношення важливості між критерієм i та критерієм j. Елементи матриці розташовані таким чином:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{1j} & \cdots & a_{ij} \end{pmatrix}, \tag{2.10}$$

де A – матриця попарних порівнянь, a_{ij} – елементи матриці парних порівнянь.

Перш за все, необхідно нормалізувати матрицю парних порівнянь, щоб забезпечити вимогу, що сума всіх елементів у кожному стовпці матриці дорівнює 1.

$$a'_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} \tag{2.11}$$

де a'_{ij} – нормалізований елемент матриці парних порівнянь, a_{ij} – початковий елемент матриці парних порівнянь.

Після нормалізації всіх елементів матриці отримуємо нормалізовану матрицю A':

$$A' = \begin{pmatrix} a'_{11} & \dots & a'_{1i} \\ \vdots & \ddots & \vdots \\ a'_{1j} & \cdots & a'_{ij} \end{pmatrix} \tag{2.12}$$

де A' – нормалізована матриця попарних порівнянь, a'_{ij} – нормалізований елемент матриці парних порівнянь.

Крок 3.2. Побудова загального вектору критеріїв та оцінка ваги векторів відносно важливості кожного критерію.

Обчислюємо вектори ваг W для кожного критерію на основі нормалізованої матриці парних порівнянь A', для подальшого аналізу їхнього впливу на загальний результат.

$$W_i = \frac{1}{n} \sum_{j=1}^{n} a'_{ij} \tag{2.13}$$

де W_i – вагового коефіцієнта для i-го критерію, a'_{ij} – нормалізований елемент матриці парних порівнянь, n – кількість критеріїв.

Для розрахунку вектору, який представляє відносну важливість кожного критерію і буде використаний для подальших розрахунків, скористаємось наступною формулою:

$$W = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{pmatrix} \tag{2.14}$$

де W – це вектори ваг критеріїв порівняння.

Крок 3.3. Аналіз узгодженості матриці попарних порівнянь.

Далі, для оцінки узгодженості матриці парних порівнянь і точності визначених вагових коефіцієнтів, необхідно розрахувати вектор *Ax:*

$$Ax = A \times W \tag{2.15}$$

де A – початкова матриця парних порівнянь, W – вектор ваг.

Отже, вектор Ax допомагає нам зрозуміти, як кожен критерій впливає на загальний результат, враховуючи відносну важливість кожного критерію.

Для перевірки узгодженості матриці парних порівнянь, що забезпечує логічну узгодженість і надійність визначених вагових коефіцієнтів, розрахуємо найбільше власне число:

$$\lambda_{\max} = \frac{1}{n} \sum_{i=1}^{n} \frac{(Ax)_i}{W_i} \tag{2.16}$$

де $\lambda_{\max}$ – найбільше власне число, n – кількість критеріїв, Ax – елементи векторів, W_i – елементи вектора ваг.

Індекс узгодженості визначає, наскільки узгодженою є матриця парних порівнянь:

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{2.17}$$

де CI – індекс узгодженості, $\lambda_{\max}$ – найбільше власне число, n – кількість критеріїв.

$$CR = \frac{CI}{RI} \tag{2.18}$$

де CR – коефіцієнт узгодженості, CI – індекс узгодженості, RI – випадковий індекс узгодженості, залежить від кількості критеріїв і визначається таблицею для відповідних значень n.

– Якщо $CR < 0{,}1$ – матриця парних порівнянь вважається узгодженою.

– Якщо $CR \geq 0{,}1$ – матриця має розбіжності і потребує перегляду парних порівнянь для досягнення кращої узгодженості.

Крок 3.4. Побудова матриць вектору критеріїв та оцінка ваги векторів для кожного елемента множини баз даних DB та множини елементів критеріїв EC.

Після того, що було встановлено, що матриця попарних порівнянь є узгодженою необхідно обчислити вектори ваг W_{EC}. Для кожного критерію на основі матриці парних порівнянь A, що необхідно для визначення відносної важливості кожного критерію та для подальшого аналізу їхнього впливу на загальний результат, необхідно обрахувати наступний вираз:

$$W_{EC} = \frac{1}{n}\sum_{j=1}^{n} a_{ij} \tag{2.19}$$

де W_{EC} – вагового коефіцієнта для *i*-го критерію, a_{ij} – елемент матриці парних порівнянь, *n* – кількість критеріїв.

Для розрахунку вектору, який представляє відносну важливість кожного критерію і буде використаний для подальших розрахунків, скористаємось наступною формулою:

$$W_{EC} = \begin{pmatrix} W_{EC1} \\ W_{EC2} \\ \vdots \\ W_{ECn} \end{pmatrix} \tag{2.20}$$

де W_{EC} – це вектори ваг критеріїв порівняння.

Крок 3.5. Визначення рангу найбільш ефективних баз даних DB.

Після того як для кожного елемента баз даних DB були побудовані матриці векторів критеріїв та оцінено ваги цих векторів необхідно обрахувати ранг найбільш ефективних баз даних DB.

$$RN_{DB} = \sum_{j=1}^{q} EC_j \cdot W_{EC}. \tag{2.21}$$

В процесі дослідження було визначено, що до переліку найбільш ефективних за рангом БД можемо відносити значення, що відповідають: $RN_{DB} \geq 0,75$.

Етап 4. Введення множини облікових задач.

У зв'язку з тим, що кожен елемент з наведеної множини баз даних DB має виконувати необхідні облікові задачі, введемо множину необхідних облікових задач у наступному вигляді:

$$\mathbf{TS} = \{\bigcup_{k=1}^{p} TS_k\} = \{TS_1, TS_2, \ldots, TS_p\}, \tag{2.22}$$

де $TS_k \subseteq \mathbf{TS}\ (k = \overline{1, p})$ – перелік облікових задач необхідних для ефективної роботи кожної СУБД, p – загальна кількість облікових задач.

Етап 5. Результат вибору найбільш ефективних баз даних.

Результати вибору найбільш ефективних БД представлені у вигляді кортежу:

$$MEF_{DB} = \langle RN_{DB}, EC_j, TS_k \rangle, \tag{2.23}$$

що характеризує найбільш пріоритетні за рангом БД RN_{DB}, які відповідають множині критеріїв EC_j, та здатні забезпечити виконання множини необхідних задач TS_k [21, 32].

Більш детальний опис етапів та кроків реалізації моделі онтологіко-реляційного сховища даних, представлено на рис. 2.2.

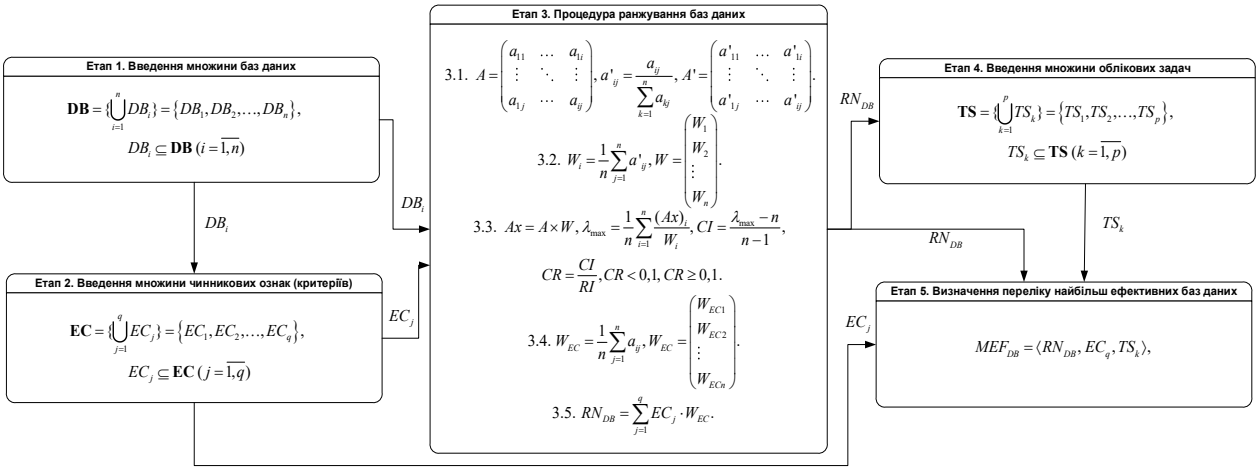


Рис. 2.2. Схема реалізації етапів моделі онтологіко-реляційного сховища даних

Таким чином, було розроблено модель онтологіко-реляційного сховища даних, яка дає можливість покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість пошуку та отримання великих обсягів інформації.

2.3. Висновки до другого розділу

Удосконалено структурно-аналітичну модель оброблення даних, яка завдяки формулюванню команд для передачі керування програмному клієнту ІКТ та додатковій обробці метаданих у хмарній системі та інтелектуалізованому виявленню сигнатур, дозволяє підвищити ефективність виявлення аномалій в хмарних ІКС.

Вперше розроблено модель онтологіко-реляційного сховища даних, яка за рахунок попередньої індексації та синтезу різних баз даних з відповідними характеристиками, дає можливість покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість пошуку та отримання великих обсягів інформації.

2.4. Список літератури до другого розділу

1. Gnatyuk S., Satybaldiyeva F., Sydorenko V., Zhyharevych O., Polozhentsev A. «Model of Information Technology for Efficient Data Processing in Cloud-based Malware Detection Systems of Critical Information Infrastructure», CEUR Workshop Proceedings, 2023, vol. 3421, pp. 206–213.

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

3. Melnyk V., Bahnyuk N., Melnyk K., Zhyharevych O., Panasyuk N. «Implementation of the simplified communication mechanism in the cloud of high performance computations», East-European Journal of Enterprise Technologies, 2017, no. 2/2(86), pp. 24–32.

4. Melnyk V., Pekh P., Melnyk K., Bahnyuk N., Zhyharevych O. «Design and implementation of interdomain communication mechanism for high performance data processing», East-European Journal of Enterprise Technologies, 2016, no. 1(9), pp. 10–15.

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

6. Mid-Year Update: 2022 SonicWall Cyber Threat Report, 39 p.

7. Aslan Ö., Ozkan-Okay M., Gupta D. «Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment», IEEE Access, vol. 9, 2021, pp. 83252–83271.

8. Xiao F., Lin Z., Sun Y., Ma Y. «Malware detection based on deep learning of behavior graphs», Mathematical Problems in Engineering, 2019.

9. Бобровнікова К.Ю., Денисюк Д.О. «Метод виявлення шкідливого програмного забезпечення шляхом аналізу мережного трафіку та поведінки програмного забезпечення в комп’ютерних системах», Вісник Хмельницького національного університету, 2020, т. 1, № 4 (287), с. 7–11.

10. Mirza Q.K.A., Awan I., Younas M. «A cloud-based energy efficient hosting model for malware detection framework», Proceedings of the IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6.

11. Zhou W., Yu B. «A cloud-assisted malware detection and suppression framework for wireless multimedia system in IoT based on dynamic differential game», China Communications, 2018, vol. 15, no. 2, pp. 209–223.

12. Indirapriyadarsini P., Mohiuddin M.U., Taqueeuddin M., Reddy C.S., Koushik T. «Malware detection using machine learning and cloud computing», International Journal of Research in Applied Sciences and Engineering Technology, 2020, vol. 8, no. 6, pp. 101–104.

13. Deyannis D., Papadogiannaki E., Kalivianakis G., Vasiliadis G., Ioannidis S. «TrustAV: Practical and privacy preserving malware analysis in the cloud», Proceedings of the 10th ACM Conference on Data and Application Security and Privacy, 2020, pp. 39–48.

14. Mondal S., Athreya D., Davies-Venn E., Zhang Z. and Aygün K. «An Improved Methodology for High Frequency Socket Performance Characterization». 2022 IEEE 31st Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), San Jose, CA, USA, 2022, pp. 1-3.

15. Rokonuzzaman M., Mishu M. K., Islam M. R., Hossain M. I., Shakeri M. and Amin N. «Design and Implementation of an IoT-Enabled Smart Plug Socket for Home Energy Management». 2021 5th International Conference on Smart Grid and Smart Cities (ICSGSC), Tokyo, Japan, 2021, pp. 50-54, doi: 10.1109/ICSGSC52434.2021.9490420.

16. Ooi R. C. et al. «High Density Interconnect (HDI) Socket Flow & Waprage Prediction & Characterization». 2022 IEEE 24th Electronics Packaging Technology Conference (EPTC), Singapore, Singapore, 2022, pp. 632-638, doi: 10.1109/EPTC56328.2022.10013256.

17. Zhang Z., Xiao W., He M., Xi J. and Mao Y. «Research and Analysis about the Length of Vertex-Degree Sequence of Complex Networks with Poisson Distribution». 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 2017, pp. 28-31, doi: 10.1109/CSE-EUC.2017.16.

18. A Poisson Process Model for Monte Carlo, in Perturbations, Optimization, and Statistics, MIT Press, 2017, pp.193-231.

19. Sharma D.K., Singh B., Raja M., Regin R. and Rajest S.S. «An Efficient Python Approach for Simulation of Poisson Distribution». 2021 7th International Conference on Advanced Computing and Communication Systems, Coimbatore, India, 2021, pp. 2011-2014.

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

21. Жигаревич О.К., Бердибаєв Р.Ш., Сидоренко В.М., Положенцев А.А., Кримська А.О. «Модель онтоголіко-реляційного сховища даних для функціонування хмарної SIEM-системи», Проблеми інформатизації та управління. 2023. Т. 4. № 76. С. 17-27.

22. Vielberth M., Pernul G. «A security information and event management pattern», Proceedings of the 12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP), 2018, pp. 1–12.

23. Agrawal K., Makwana H. «A study on critical capabilities for security information and event management», International Journal of Science and Research (IJSR), 2015, vol. 4, no. 7, pp. 1893–1896.

24. Ribolovlev D., Karasov S., Polyakov S. «Classification of emergency management systems for incidents without backing», Food of Cyber Security, 2018, no. 3(27), pp. 47–53.

25. Lee J., Kim Y., Kim J., Kim I. «Toward the SIEM architecture for cloud-based security services», 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, 2017, pp. 398–399. DOI: 10.1109/CNS.2017.8228696.

26. Bachane I., Adsi Y.I.K., Adsi H.C. «Real time monitoring of security events for forensic purposes in cloud environments using SIEM», 2016 Third International Conference on Systems of Collaboration (SysCo), 2016, pp. 1–3. DOI: 10.1109/SYSCO.2016.7831327.

27. AlSabbagh B., Kowalski S. «A framework and prototype for a socio-technical security information and event management system (ST-SIEM)», 2016

European Intelligence and Security Informatics Conference (EISIC), 2016, pp. 192–195. DOI: 10.1109/EISIC.2016.049.

28. Serckumecka A., Medeiros I., Bessani A. «Low-cost serverless SIEM in the cloud», 2019 38th Symposium on Reliable Distributed Systems (SRDS), 2019, pp. 381–391. DOI: 10.1109/SRDS47363.2019.00057.

29. Nabil M., Soukainat S., Lakbabi A., Ghizlane O. «SIEM selection criteria for an efficient contextual security», 2017 International Symposium on Networks, Computers and Communications (ISNCC), 2017, pp. 1–6. DOI: 10.1109/ISNCC.2017.8072035.

30. Mahmoud R.-V., Kidmose E., Turkmen A., Pilawka O., Pedersen J.M. «DefAtt – architecture of virtual cyber labs for research and education», 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2021, pp. 1–7.

31. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. «A concept of the architecture and creation for SIEM system in critical infrastructure». Studies in Systems, Decision and Control. 2021. Vol. 346. P. 221–242.

32. Sydorenko V., Zhyharevych O., Berdybaev R., Polozhentsev A., Fesenko A. «Ontological-Relational Data Store Model for a Cloud-based SIEM System Development». CEUR Workshop Proceedings. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2024), February 28, 2024, Kyiv, Ukraine, 2024. Vol. 3654. P. 343-354.

33. Gnatyuk S., Berdibayev R., Aleksander M., Sydorenko V., Zhyharevych O., Polozhentsev A. «Software System for Cybersecurity Events Correlation and Incident Management in Critical Infrastructure». Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies. 2024. Vol. 213. P. 247-269. Springer, Cham.

34. Polozhentsev A., Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O. «Novel Cyber Incident Management System for 5G-based Critical Infrastructures». IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, P. 1037-1041.

РОЗДІЛ 3
РОЗРОБКА СИСТЕМИ КОРЕЛЮВАННЯ ПОДІЙ ТА УПРАВЛІННЯ ІТ-ІНЦИДЕНТАМИ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

3.1. Модель інтеграційної шини даних для ефективного функціонування системи управління подіями інформаційної безпеки

Запропонована в роботах [1-3] модель інтеграційної шини даних (ІШД), призначена для забезпечення ефективного функціонування системи управління подіями інформаційної безпеки, розроблена з урахуванням результатів проведеного аналізу сучасних підходів до побудови інтеграційних рішень [4-7].

Особливості архітектури сучасних ІШД

Основою запропонованої моделі є використання архітектурного підходу Service-Oriented Architecture (SOA), у межах якого ІШД виступає центральним компонентом взаємодії між сервісами. SOA забезпечує можливість багаторазового використання програмних компонентів за рахунок реалізації сервісних інтерфейсів (рис. 3.1) [8]. Такі інтерфейси базуються на уніфікованих комунікаційних стандартах, що дозволяє забезпечити їх швидку інтеграцію у нові або існуючі ІС без необхідності виконання складної повторної інтеграції.

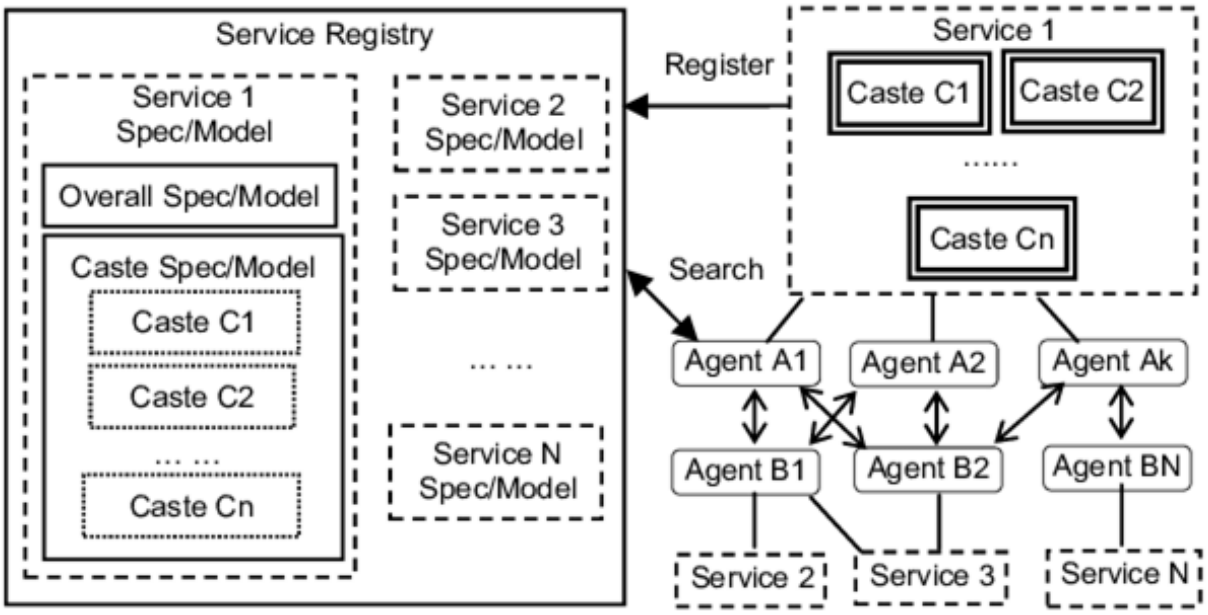


Рис. 3.1. Схема побудови SOA

SOA-сервіс включає програмний код та механізми інтеграції даних, необхідні для виконання визначеної функції [9]. Низький рівень взаємозалежності сервісів забезпечує можливість їх повторного використання та спрощує інтеграцію.

Доступ до сервісів здійснюється через стандартні мережеві протоколи (SOAP/HTTP, JSON/HTTP), що забезпечують уніфікований обмін даними. Сервіси можуть бути як розроблені окремо, так і інтегровані на основі існуючих систем шляхом експорту відповідних інтерфейсів.

У межах SOA взаємодія сервісів здійснюється незалежно від їх реалізації або платформи. Обмін даними між сервісами реалізується через ІШД [10], яка виступає основним елементом SOA-архітектури. Таким чином, ІШД виступає шаблоном (рис. 3.2), що забезпечує централізовану інтеграцію систем, маршрутизацію запитів та доступ до сервісів через стандартизовані інтерфейси.

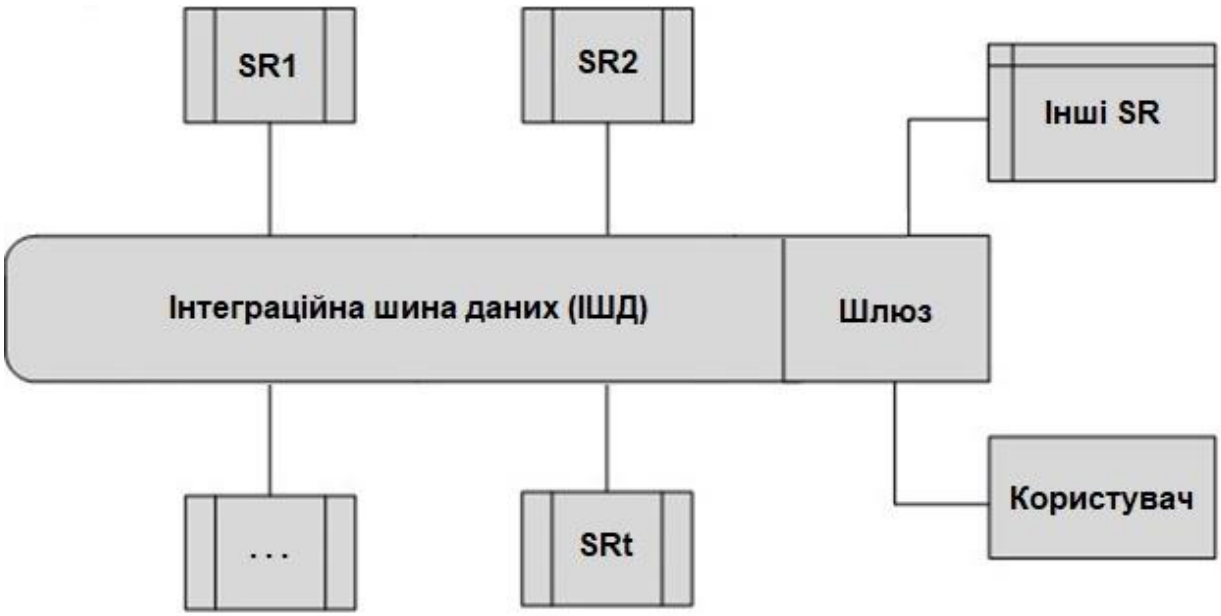


Рис. 3.2. Схема побудови ІШД

ІШД забезпечує перетворення моделей даних, маршрутизацію повідомлень та взаємодію сервісів, об’єднуючи ці функції в єдиному сервісному інтерфейсі, що може багаторазово використовуватися різними додатками. Реалізація ІШД здійснюється за допомогою спеціалізованих інтеграційних платформ та інструментів [11].

Застосування ІШД дозволяє уникнути складної взаємодії типу point-to-point, зменшити кількість залежностей між сервісами та спростити подальше обслуговування системи (рис. 3.3).

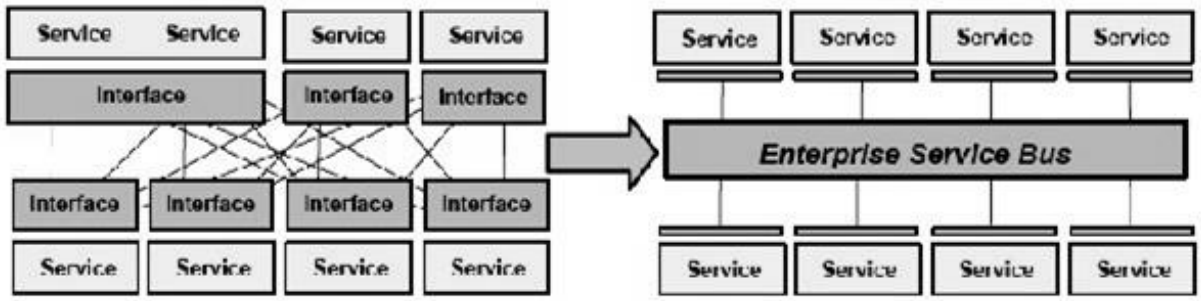


Рис. 3.3. Порівняння SOA з ІШД та без нього

ІШД може бути реалізована з використанням брокерів повідомлень та стандартних форматів обміну даними, таких як XML або JSON. При цьому взаємодія між сервісами здійснюється через адаптери, що забезпечують перетворення даних у необхідний формат. На рис. 3.4 наведено приклад реалізації ІШД, який демонструє підхід до інтеграції сервісів із використанням механізмів трансформації та маршрутизації повідомлень [1].

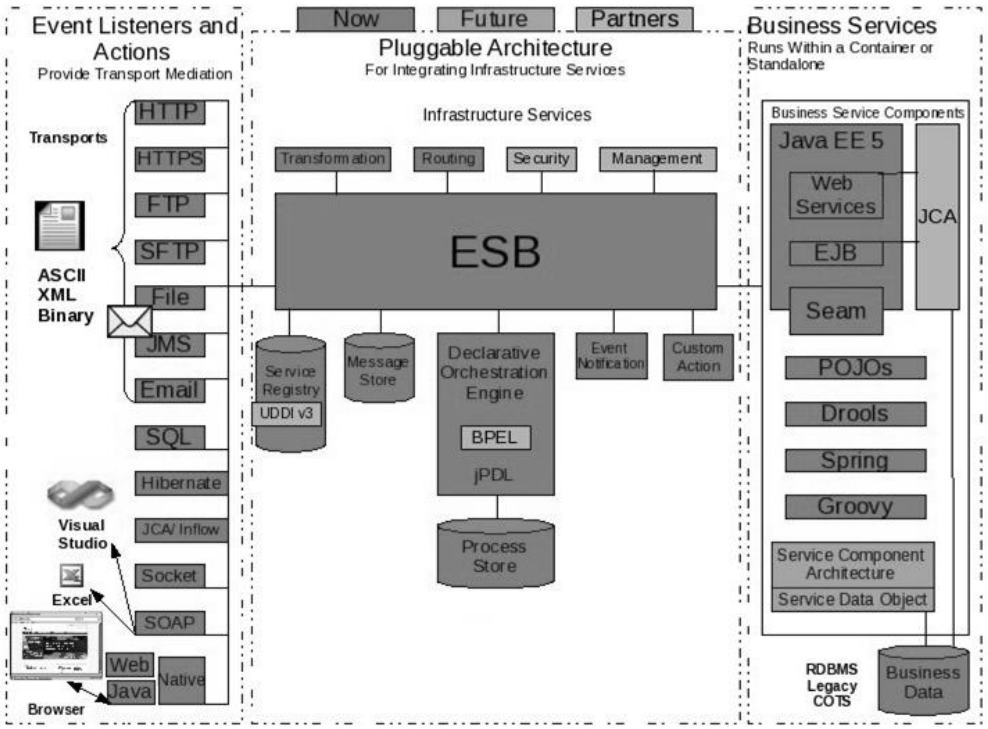


Рис. 3.4. Приклад реалізації JBoss ІШД

Впровадження ІШД для ефективного функціонування SIEM-систем

Розроблена платформа [1, 4, 12] використовує механізм кореляції подій безпеки, що дозволяє віднести її до повнофункціональних SIEM-систем. Платформа забезпечує кореляцію нормалізованих подій, виконання запитів для аналізу загроз і джерел вразливостей, а також генерацію попереджень з урахуванням рівня ризику [13].

В умовах цифрової трансформації організації використовують значну кількість ІС, які оперують взаємопов'язаними даними. Використання ІШД дозволяє організувати обмін інформацією між такими системами без необхідності їх модифікації, забезпечуючи узгоджену взаємодію сервісів та збалансований розподіл навантаження. У традиційній архітектурі типу point-to-point взаємодія між сервісами призводить до зростання кількості зв'язків, що ускладнює підтримку системи та знижує її надійність (рис. 3.5).

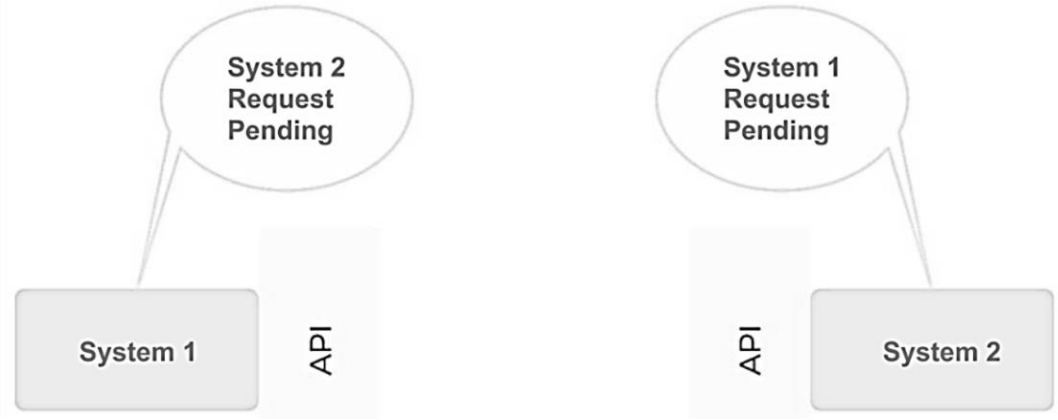


Рис. 3.5. Взаємодія служб без шини

Використання ІШД дозволяє централізувати взаємодію сервісів, усуваючи необхідність прямого обміну між ними. У цьому випадку всі компоненти системи взаємодіють через інтеграційну платформу, що підвищує гнучкість, спрощує масштабування та зменшує вплив змін окремих підсистем на роботу всієї системи (рис. 3.6).

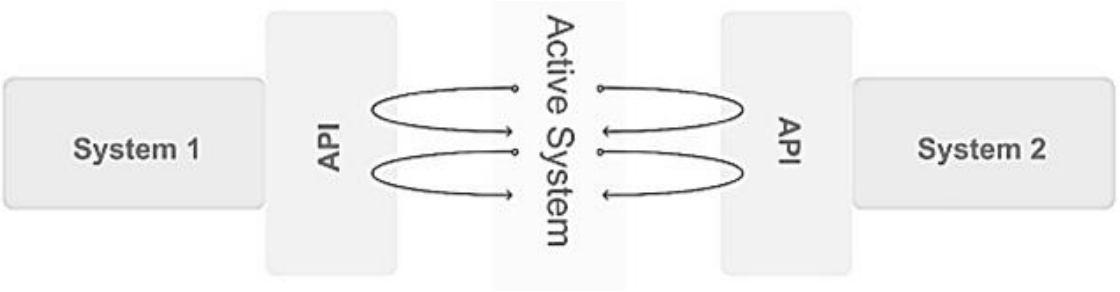


Рис. 3.6. Взаємодія служб з шиною

Таким чином, застосування ІШД забезпечує підвищення ефективності функціонування системи за рахунок зменшення кількості залежностей між сервісами, спрощення інтеграції та підвищення надійності обміну даними.

Для збору подій система використовує агентів, що встановлюються у контрольованих підсистемах, а також стандартні механізми (syslog, SNMP). Для аналізу мережевого трафіку може використовуватися NetFlow.

У розробленій системі для забезпечення обміну даними використовується брокер повідомлень, який забезпечує гарантовану доставку повідомлень від декількох джерел до споживачів. Репозиторій системи виконує функцію зберігання необроблених даних у захищеному вигляді, що дозволяє використовувати їх для подальшого аналізу та розслідування інцидентів.

Архітектурною особливістю платформи є використання горизонтально масштабованих розподілених баз даних [14], що забезпечує:

- високу швидкість обробки великих обсягів даних;
- мінімальні затримки;
- відмовостійкість;
- можливість розширення без зупинки системи.

У системі реалізовано такі функціональні модулі:

- *модуль моніторингу,* який забезпечує контроль метрик у реальному часі та формування подій безпеки;
- *модуль аналітики,* що виконує нормалізацію, кореляцію та аналіз подій, а також ідентифікацію інцидентів.

Інтеграція компонентів системи здійснюється через API [15], що забезпечує стандартизовану взаємодію між сервісами (рис. 3.7).

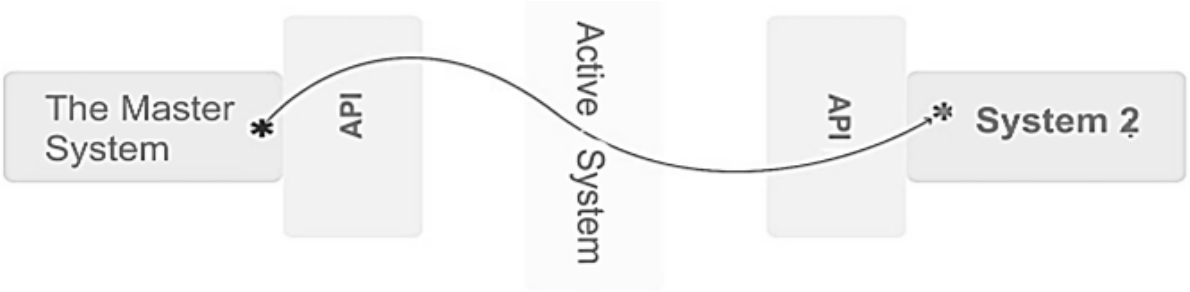


Рис. 3.7. Взаємодія SOA-сервісу з шиною

Встановлено, що для ефективного функціонування системи корелювання подій та управління IT-інцидентами на ОКІ необхідно забезпечити безперервне надання сервісів SIEM-систем. Під сервісами у роботі розуміються послуги, що надаються SIEM-системами. А ІШД, в свою чергу, має систематизувати та інтегрувати роботу таких сервісів. Основним завданням ІШД є визначення режимів функціонування сервісів і формування оптимального порядку їх оброблення з урахуванням рівня критичності, що дає змогу мінімізувати втрати часу на очікування обслуговування та простої каналів. Для цього черга обслуговування сервісів визначається за рівнем їх критичності з використанням підходу FMECA [18-20].

Етап 1. Введення множини сервісів системи.

Введемо множину сервісів системи **SR** у наступному вигляді:

$$\mathbf{SR} = \{\bigcup_{s=1}^{t} SR_s\} = \{SR_1, SR_2, \ldots, SR_t\}, \tag{3.1}$$

де $SR_s \subseteq \mathbf{SR}$ $(s = \overline{1,t})$ – різновиди сервісів системи, які надають послуги SIEM-системам, t – загальна кількість сервісів.

Етап 2. Визначення критичності сервісів.

Для оцінювання критичності сервісів введемо множину значень рангу критичності **RN** у наступному вигляді:

$$\mathbf{RN} = \{\bigcup_{r=1}^{w} RN_r\} = \{RN_1, RN_2, \ldots, RN_w\}, \tag{3.2}$$

де $RN_r \subseteq \mathbf{RN}\ (r=\overline{1,w})$ – ранги критичності сервісів SR, де w – загальна кількість показників рангів критичності.

Ранг критичності сервісів SR визначається інтегральною оцінкою [16-20]:

$$RN_r = IN_{1a} \cdot IN_{2a} \cdot IN_{3a} \tag{3.3}$$

Крок 2.1. Для визначення показника IN_{1a} (оцінка ймовірності настання переривання роботи сервісу SR) введемо відповідну множину: $\mathbf{IN}_1 = \{\bigcup_{a=1}^{z} IN_{1a}\} = \{IN_{11}, IN_{12}, \ldots, IN_{1z}\}$, де $IN_{1a} \subseteq \mathbf{IN}_1\ (a=\overline{1,z})$ – значення z знаходяться за таблицею, сформованою у залежності від типу SIEM-систем.

Для визначення показника IN_{1a} значення z знаходимо згідно [16-18], відповідно до табл. 3.1.

Таблиця 3.1

Значення коефіцієнта IN_1

Категорія частоти виникнення події	Інтервал інтенсивності події	Значення IN_1 бали
Дуже рідкісний	$< 2\cdot10^{-7}$	1
	$2\cdot10^{-7} \ldots 8\cdot10^{-6}$	2
Рідкісний	$8\cdot10^{-6} \ldots 2\cdot10^{-6}$	3
	$2\cdot10^{-6} \ldots 8\cdot10^{-5}$	4
Можливий	$8\cdot10^{-5} \ldots 2\cdot10^{-5}$	5
	$2\cdot10^{-5} \ldots 8\cdot10^{-4}$	6
Частковий	$8\cdot10^{-4} \ldots 2\cdot10^{-4}$	7
	$2\cdot10^{-4} \ldots 5\cdot10^{-3}$	8
Дуже частковий	$5\cdot10^{-3} \ldots 2\cdot10^{-3}$	9
	$> 2\cdot10^{-3}$	10

Крок 2.2. Для визначення показника IN_{2a} (цінка ймовірності попереднього виявлення переривання сервісу) введемо відповідну множину: $\mathbf{IN}_2 = \{\bigcup_{a=1}^{x} IN_{2a}\} = \{IN_{21}, IN_{22}, \ldots, IN_{2x}\}$, де $IN_{2a} \subseteq \mathbf{IN}_2\ (a=\overline{1,x})$ – значення x знаходяться за таблицею, сформованою у залежності від типу SIEM-систем.

Для визначення показника IN_{2a} значення x знаходимо згідно [16-18], відповідно до табл. 3.2.

Таблиця 3.2

Значення коефіцієнта IN_2

Категорія виявлення події	Опис умов	Значення IN_2 бали
Дуже висока	Переривання сервісу гарантовано виявляється до впливу на користувача	1
Висока	Переривання виявляється на етапі внутрішнього контролю або тестування	2
Достатня	Переривання може бути виявлене під час експлуатаційного контролю	3
Середня	Переривання виявляється після часткового впливу на систему	4
Помірна	Переривання може бути пропущене на етапі тестування	5
Знижена	Виявлення можливе лише під час експлуатації або після інциденту	6
Низька	Висока ймовірність невиявлення переривання на ранніх етапах	7
Дуже низька	Переривання майже не виявляється стандартними засобами контролю	8
Критично низька	Переривання виявляється лише після значних наслідків	9
Відсутня	Переривання не піддається виявленню існуючими засобами	10

Крок 2.3. Для визначення показника IN_{3a} (оцінка тяжкості настання переривання роботи сервісу) введемо множину: $\mathbf{IN}_3 = \{\bigcup_{a=1}^{c} IN_{3a}\} = \{IN_{31}, IN_{32}, \ldots, IN_{3c}\}$, де $IN_{3a} \subseteq \mathbf{IN}_3$ $(a = \overline{1,c})$ – значення c знаходяться за таблицею, сформованою у залежності від типу SIEM-систем.

Для визначення показника IN_{3a} значення c знаходимо згідно [16-18], відповідно до табл. 3.3.

Крок 2.4. Розрахунок рангу критичності RN_r для кожного з перерахованих видів сервісів SIEM-систем згідно до (3.3).

Таблиця 3.3

Значення коефіцієнта IN_3

Рівень наслідків	Характеристика впливу	Значення IN_3 бали
Мінімальний	Переривання не впливає на функціонування системи та майже непомітне для користувача	1
Дуже низький	Незначні відхилення, що виявляються під час тестування та не впливають на роботу системи	2
Низький	Локальні порушення, що не впливають на основні функції системи	3
Помірний	Часткове зниження продуктивності або доступності окремих сервісів	4
Суттєвий	Вплив на окремі функціональні компоненти системи, що потребує втручання	5
Середній	Порушення роботи сервісів, що призводить до обмеження функціональності системи	6
Високий	Значне порушення роботи системи та зниження ефективності її функціонування	7
Дуже високий	Втрата працездатності ключових сервісів або підсистем	8
Критичний	Повна недоступність системи або значної її частини	9
Катастрофічний	Порушення роботи системи, що призводить до серйозних наслідків для безпеки критичної інфраструктури	10

Етап 3. Виділення переліку критичних сервісів SIEM-систем.

Введемо правила для визначення критичності сервісу $crt(SR_s) \in \{H, M, L\}$:

$$crt(SR_s) = \begin{cases} H, if\ RN_r > RN_k; \\ M, if\ RN_0 < RN_r \le RN_k; \\ L, if\ RN_r \le RN_0, \end{cases} \tag{3.4}$$

де RN_0, RN_k – порогові значення рангу критичності. У випадку якщо $crt(SR_s) = H$ робота сервісу визнається критичним, при $crt(SR_s) = M$ необхідне застосування заходів щодо зменшення критичності, при $crt(SR_s) = L$

переривання роботи сервісу вважається незначним і не потребує впровадження додаткових заходів.

Етап 4. Ранжування переліку критичних сервісів SIEM-систем за допомогою діаграми Парето.

На цьому етапі для виділення переліку найбільш значущих (критичних) сервісів використовується стовпчаста діаграма Парето, [16] (рис. 3.8), яка будується окремо для кожної SIEM-системи (з метою ранжування найбільш значущих (критичних) сервісів по горизонтальній осі діаграми відкладаються сервіси SR_s, а по вертикальній – розраховане значення RN_r, якщо $RN_r > RN_k$, то сервіси на діаграмі позначається червоним кольором, якщо $RN_0 < RN_r \le RN_k$, то сервіси на діаграмі позначається жовтим кольором, якщо $RN_r \le RN_0$, то сервіси на діаграмі позначається зеленим кольором.

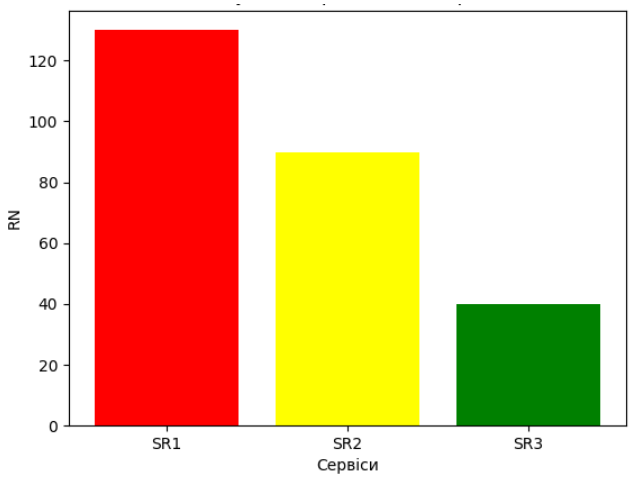


Рис. 3.8. Приклад побудови стовпчастої діаграми Парето

Діаграма Парето дозволяє визначити перелік найбільш значущих (критичних) сервісів у межах однієї SIEM-системи, а також забезпечує можливість порівняння критичних сервісів між різними SIEM-системами. Крім того, її використання дає змогу здійснювати ранжування сервісів за рівнем критичності з метою оптимального розподілу навантаження в системі.

Отже, запропонована модель інтеграційної шини даних дозволяє ефективно розподіляти навантаження між сервісами та забезпечує безперервність обміну даними, що є необхідною умовою функціонування

системи корелювання подій та управління ІТ-інцидентами. Застосування розробленої моделі ІШД у SIEM-системах забезпечує такі переваги: масштабованість рішення; гнучку маршрутизацію потоків даних; гарантовану доставку повідомлень; організацію захищених каналів передачі; централізоване управління сервісами; можливість моніторингу та діагностики процесів обміну даними; інтеграцію зі сторонніми системами та брокерами повідомлень.

3.2. Система корелювання подій та управління ІТ-інцидентами на ОКІ

На сьогодні системи управління інформацією та подіями ІБ (SIEM, Security Information and Event Management) є одним із ключових напрямів розвитку засобів захисту, що забезпечують ефективне виявлення загроз та формування заходів протидії для підтримання належного рівня захисту інформаційної інфраструктури.

Функціонування SIEM-систем ґрунтується на оперативному збиранні, зберіганні та аналітичному обробленні подій безпеки, які формуються в журналах різних апаратних і програмних компонентів інформаційної інфраструктури, зокрема серверів, робочих станцій, мережевого обладнання, систем виявлення атак та інших засобів захисту [22-28].

Основною метою SIEM-систем є підвищення рівня ІБ в ІКС за рахунок оброблення та аналізу подій безпеки в режимі, близькому до реального часу, а також реалізації проактивного управління інцидентами. Проактивний підхід передбачає виконання дій до настання критичних ситуацій і базується на використанні історичних даних, прогнозуванні загроз та адаптивному налаштуванні параметрів моніторингу відповідно до поточного стану системи [28-30].

Для досягнення зазначеної мети SIEM-системи, що застосовуються на ОКІ, повинні забезпечувати виконання таких основних завдань:

- збирання, оброблення та аналіз подій безпеки з різнорідних джерел;
- виявлення атак і порушень політик безпеки в режимі, близькому до реального часу;
- оцінювання рівня захищеності КВР;

- аналіз та управління ризиками інформаційної безпеки;
- проведення розслідувань інцидентів;
- виявлення невідповідностей між ресурсами та політиками безпеки;
- підтримка прийняття рішень;
- формування звітності [2, 38].

Основними вхідними даними для SIEM-систем є записи журналів подій (logs), що відображають дії користувачів і програм, які можуть впливати на безпеку системи. Із загального потоку подій необхідно ідентифікувати ті, що свідчать про атаки або інші небажані дії, при цьому традиційні підходи до їх аналізу є трудомісткими та недостатньо ефективними.

Типова архітектура SIEM-системи включає три основні компоненти: агенти, сховище даних і сервер оброблення. Агенти забезпечують збір та первинну обробку подій, сховище даних виконує функції їх зберігання, а сервер оброблення здійснює аналітичну обробку, формування попереджень і підтримку прийняття управлінських рішень.

Таким чином, архітектура SIEM-системи може бути представлена у вигляді трьох основних рівнів: *збирання даних, управління даними* та *аналіз даних.*

На рівні *збирання даних* здійснюється отримання інформації з різнорідних джерел, зокрема серверів, мережевого обладнання, систем захисту та інших компонентів інформаційної інфраструктури.

Рівень *управління даними* забезпечує зберігання, структуризацію та надання доступу до даних для подальшої обробки.

На рівні *аналізу даних* виконується кореляція подій, формування звітів та генерація попереджень у режимі, близькому до реального часу.

SIEM-система поєднує функції двох класів систем управління інформаційною безпекою – SIM (Security Information Management) та SEM (Security Event Management) [2, 34-39]. SIM забезпечує збирання, зберігання та аналіз журналів подій і формування звітності, тоді як SEM орієнтована на моніторинг подій у режимі, близькому до реального часу, а також виявлення та реагування на інциденти безпеки.

Реалізація зазначених функцій передбачає використання комплексу механізмів оброблення подій безпеки, зокрема нормалізації, фільтрації, класифікації, агрегації, кореляції та пріоритезації подій, а також формування звітів і попереджень. У SIEM-системах нового покоління додатково застосовуються механізми аналізу інцидентів, оцінювання їх наслідків, підтримки прийняття рішень і візуалізації результатів [31].

Зазначені механізми забезпечують комплексну обробку подій безпеки – від їх первинного збирання до формування управлінських рішень щодо реагування на інциденти.

Взаємозв'язок механізмів оброблення подій у SIEM-системах відображається у функціональній моделі, що включає п'ять основних підсистем: збирання, оброблення, зберігання, аналіз та представлення даних. При цьому підсистеми збирання та оброблення функціонують у режимі реального часу, тоді як інші — у режимі, близькому до реального часу.

1) Підсистема збирання даних забезпечує отримання інформації з різнорідних джерел із використанням методів Push і Pull.

2) Підсистема оброблення виконує нормалізацію, фільтрацію, агрегацію, класифікацію та кореляцію подій.

3) Підсистема зберігання відповідає за накопичення та структуризацію даних у відповідних сховищах, тоді як підсистема аналізу реалізує кореляцію, пріоритезацію подій, аналіз інцидентів і підтримку прийняття рішень.

4) Підсистема аналізу може базуватися як на кількісних, так і на якісних оцінках залежно від вимог до швидкодії та точності.

5) Підсистема представлення забезпечує візуалізацію результатів, формування звітів та генерацію попереджень [38-39].

Отже, все вище зазначене свідчить про необхідність розроблення універсальної системи корелювання подій та управління IT-інцидентами на ОКІ, яка враховує зазначені функціональні вимоги.

При цьому для забезпечення відповідності вимогам ІБ така система повинна відповідати міжнародним стандартам, зокрема ISO/IEC 27000, PCI DSS, HIPAA, NIST SP 800-171, DoD RMF та GDPR.

У роботах [2, 21, 38-39] запропоновано та досліджено систему корелювання подій та управління IT-інцидентами на ОКІ. У процесі дослідження [22-30] встановлено, що на сучасному етапі доцільним є використання відкритих (Open Source) рішень з огляду на зниження витрат і можливість адаптації функціоналу відповідно до потреб конкретного ОКІ. Водночас, з позицій ІБ, найбільш ефективним є підхід, що передбачає розроблення власної системи корелювання подій та управління IT-інцидентами, яка забезпечує розширений функціонал, гнучкість, масштабованість та захищеність від потенційних уразливостей і несанкціонованих механізмів доступу.

На основі цього запропоновано універсальне рішення, що базується на використанні хмарної SIEM-системи для КІ [5], а також розроблено концепцію архітектури системи корелювання подій та управління IT-інцидентами на ОКІ. Архітектуру запропонованої системи наведено на рис. 3.9.

Запропонована архітектура орієнтована на використання в різних секторах КІ та передбачає можливість інтеграції з існуючими SIEM-системами й іншими інструментами управління інцидентами.

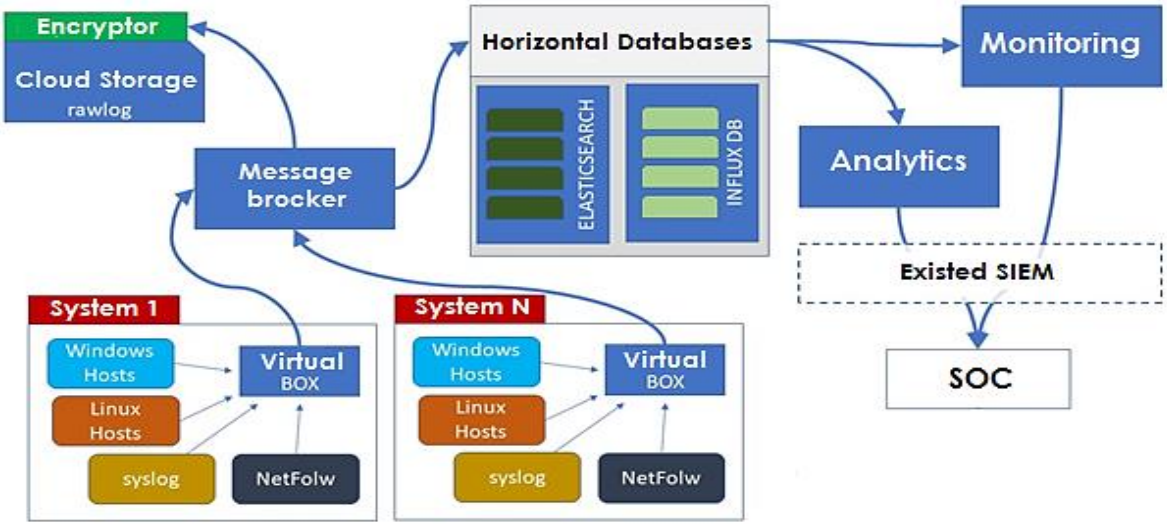


Рис. 3.9. Архітектура розробленої системи корелювання подій та управління IT-інцидентами на ОКІ

Основними структурними компонентами системи корелювання подій та управління IT-інцидентами на ОКІ є:

- горизонтальні бази даних (Horizontal Databases);
- блок аналітики (Analytics);
- блок моніторингу (Monitoring);
- хмарне сховище (Cloud Storage);
- шифратор даних (Encryptor);
- брокер повідомлень (Message Broker);
- джерела (System 1 – System *N*).

Для зазначеної системи відповідно до [2] розроблено моделі функціонування гібридного сховища даних безпеки, які відрізняються від існуючих аналогів поєднанням двох типів баз даних. Зокрема, для обробки журналів використовується масштабований повнотекстовий пошуковий рушій Elasticsearch, а для зберігання структурованих даних — документоорієнтована СУБД MongoDB. Такий підхід забезпечує ефективну індексацію даних, можливість роботи з різними типами запитів (простими, складними та структурованими), підтримку різних форматів даних, а також виконання операцій агрегації та аналітичної обробки. Крім того, система забезпечує високу швидкість пошуку та виявлення закономірностей у великих обсягах даних. Запропоноване рішення підтримує горизонтальне масштабування за рахунок кластеризації, реплікацію даних між вузлами та балансування навантаження. Це дозволяє забезпечити відмовостійкість системи, а також можливість її використання як розподіленого сховища даних.

Розроблено модель та алгоритми функціонування ІШД, які відрізняються від існуючих аналогів використанням комбінованого підходу до збору подій безпеки. Зокрема, для збору інформації застосовуються як власні агенти, що встановлюються у контрольованих системах, так і стандартні механізми збору подій (syslog, SNMP тощо), для яких передбачено сценарії інтеграції з можливістю модифікації при мінімальному втручанні розробників. Крім того, ІШД може використовуватися для моніторингу мережі як колектор

статистики NetFlow, що надходить з мережевого обладнання, а також для аналізу мережевого трафіку як шляхом отримання дзеркального трафіку, так і шляхом пропускання трафіку через систему. Запропонований підхід забезпечує високу швидкість обробки великих обсягів даних, мінімальні затримки при їх обробленні та формуванні аналітичних запитів і звітів, підвищену відмовостійкість, а також гнучкість і масштабованість системи за рахунок можливості розширення шляхом додавання нових вузлів без зупинки роботи.

З погляду інформаційної безпеки важливу роль у запропонованій системі відіграє модуль шифрування даних (Encryptor), який функціонально інтегрований із хмарним сховищем (Cloud Storage) та забезпечує конфіденційність необроблених записів після їх збору агентами (syslog, NetFlow тощо). Передача зібраних даних здійснюється у зашифрованому вигляді через брокер повідомлень (Message Broker) до горизонтально масштабованих баз даних (Horizontal Databases). У разі втрати зв’язку з брокером повідомлень передбачено механізм тимчасового зберігання даних у хмарному сховищі з подальшою передачею після відновлення з’єднання.

Розроблену систему корелювання подій та управління IT-інцидентами на ОКІ реалізовано програмно. Наступним етапом є проведення експериментального дослідження створеного програмного рішення як інструменту ІБ з метою оцінювання його відповідності визначеним критеріям та ефективності корелювання подій і управління інцидентами, що виникають у КІ та впливають на КВР.

3.3. Висновки до третього розділу

Удосконалено модель інтеграційної шини даних, яка за рахунок декомпозиції функціональності сервісів (кожен з яких відповідає за окреме завдання і може працювати ізольовано від інших) та визначення критичності сервісів, дозволяє розподілити навантаження на послуги та гарантує безперервність обміну даними для ефективного функціонування системи корелювання подій та управління IT-інцидентами.

Отримала подальший розвиток система корелювання подій та управління ІТ-інцидентами, яка за рахунок використання розроблених моделей обробки даних, онтологіко-реляційного сховища даних та інтеграційної шини даних, дає змогу формалізувати інформаційну технологію, що реалізує процеси управління ІТ-інцидентами на ОКІ згідно вимог міжнародних стандартів та найкращих світових практик щодо створення систем управління ІТ-інцидентами.

3.4. Список використаних джерел у третьому розділі

1. Гнатюк С.О., Бердибаєв Р.Ш., Богун А.М., Сидоренко В.М., Положенцев А.А., Жигаревич О.К. «Інтеграційна шина даних для ефективного функціонування системи управління подіями інформаційної безпеки», Проблеми інформатизації та управління. 2023. Т. 3. № 75. С. 29-40.

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

3. Здолбіцька Н.В., Ліщина Н.М., Лавренчук С.В., Давиденко Н.В., Жигаревич О.К. «Інтелектуальна інформаційна система «робот-гід»». Матеріали Міжнародної наукової молодіжної школи «Системи та засоби штучного інтелекту» 28.11.2021р. Київ, 2021. С. 19-21.

4. Gnatyuk S., Berdibayev R., Sydorenko V., Polozhentsev A., Ryabyy M. «Enterprise Service Bus Construction in SOA Architecture for SIEM Implementation in Critical Information Infrastructure» CEUR Workshop Proceedings, 2022, Vol. 3288, Paper 2, P. 11-20.

5. Berdibayev R., Gnatyuk S., Tynymbayev S., Sydorenko V. «Advanced technologies of cyber incident management in critical infrastructure». Kyiv: Pro Format, 2022, 125 p.

6. Sreemathy J., Joseph V.I., Nisha S., Prabha I.C., Priya R.M.G. «Data integration in ETL using Talend», 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1444-1448.

7. Mkhwanazi X., Le H., Blake E. «Clustering between data mules for better message delivery», 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 209-214.

8. Jin Z. and Zhu H. «A Framework for Agent-Based Service-Oriented Modelling». 2008 IEEE International Symposium on Service-Oriented System Engineering, 2008, pp. 160-165.

9. Li W. «Design and Implementation of Software Testing Platform for SOA-Based System». 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), 2021, pp. 1094-1098.

10. ESB (Enterprise Service Bus), https://www.ibm.com/cloud/learn/esb

11. Dai P. «Design and implementation of ESB based on SOA in power system». 2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011, pp. 519-522.

12. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. «A concept of the architecture and creation for SIEM system in critical infrastructure». Studies in Systems, Decision and Control. 2021. Vol. 346. P. 221-242.

13. Laue T., Kleiner C., Detken K.-O. and Klecker T. «A SIEM Architecture for Multidimensional Anomaly Detection». 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 136-142.

14. Asef P., Taheri R., Shojafar M., Mporas I. and Tafazolli R. «SIEMS: A Secure Intelligent Energy Management System for Industrial IoT applications». EEE Transactions on Industrial Informatics. 2023. T. 19, № 1. P. 1039-1050. DOI: 10.1109/TII.2022.3165890.

15. Orsós M., Kecskés M., Kail E. and Bánáti A. «Log collection and SIEM for 5G SOC». 2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2022, pp. 000147-000152.

16. Щербак Л., Гнатюк С., Сидоренко В., Шаховал О. «Метод визначення рівня важливості об’єктів критичної інформаційної інфраструктури в галузі цивільної авіації», Безпека інформації, 2017. Т. 23, №1, С. 27-38.

17. ISO/IEC. «Risk management — Risk assessment techniques». ISO/IEC 31010:2019, Geneva, 2019.

18. IEC. «Failure modes and effects analysis (FMEA and FMECA)». IEC 60812:2018, Geneva, 2018.

19. Stamatis D.H. «Failure Mode and Effect Analysis: FMEA from Theory to Execution». ASQ Quality Press, 2003.

20. McDermott R., Mikulak R., Beauregard M. «The Basics of FMEA». CRC Press, 2009.

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

22. Buriachok V., Sokolov V., Skladannyi P. «Security rating metrics for distributed wireless systems», IEEE International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2019, pp. 612–616. DOI: 10.1109/PICST47496.2019.9061357.

23. Kipchuk F. et al. «Investigation of availability of wireless access points based on embedded systems», 2019 IEEE International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2019. DOI: 10.1109/PICST47496.2019.9061551.

24. Жигаревич О.К., Медведєв М.В. «Інформаційна ситема “Студент - ФКНІТ” засобами PHP». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2017. № 26. С. 88-92.

25. Жигаревич О.К., Котлярець В.В., Луць А.Р. «Модель екосистеми навчального програмного забезпечення». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2017. № 26. 167-177.

26. Жигаревич О.К., Мельник В.М., Мельник К.В. «Підтримка оголошеної/встановленої комунікації в мережі через стандартні сокети API». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2015. № 19. С. 23-27.

27. Мельник К.В., Мельник В.М., Багнюк Н.В., Жигаревич О.К., Климюк М. «Система попереднього відбору кандидатів на основі нечіткої логіки». Комп’терно-інтегровані технології: освіта, наука, виробництво. 2015. № 19. С. 114-120.

28. Дмитрук Я.В., Гришанович Т.О., Глинчук Л.Я., Жигаревич О.К. «Кібервійна як різновид інформаційних війн. Захист кіберпростору України». Кібербезпека: освіта, наука, техніка. 2022. Т. 4. №16. С. 28-36.

29. Сидоренко В., Положенцев А., Юдін О., Жигаревич О. «Функціональна модель визначення критичності галузевих інформаційно-телекомунікаційних систем». ABIA-2023: XVI міжнар. наук.-техніч. конф., 18-20 квітня 2023 р.: тези доп., Київ: НАУ, 2023. С. 16.14-16.17.

30. Gnatyuk S., Sydorenko V., Yudin O., Zhyharevych O., Polozhentsev A. «Method for Calculating the Criticality Level of Sectoral Information and Telecommunication Systems». CEUR Workshop Proceedings, Proceedings of the: Information Technology and Implementation (IT&I2022), November 30 - December 02, 2022, Kyiv, Ukraine, Vol. 3347, Paper 20, P.234-245.

31. Bogachuk I., Sokolov V., Buriachok V. «Monitoring subsystem for wireless systems based on miniature spectrum analyzers», Proceedings of the International Scientific-Practical Conference on Problems of Infocommunications Science and Technology (PIC S&T), 2018, pp. 581-585. DOI: 10.1109/infocommst.2018.8632151.

32. Gnatyuk S., Berdibayev R., Fesenko A., Kyryliuk O., Bessalov A. «Modern SIEM Analysis and Critical Requirements Definition in the Context of Information Warfare», CEUR Workshop Proceedings, 2021, vol. 3188, pp. 149-166.

33. Miller D., Harris S., Harper A., VanDyke S., Blask C. «Security information and event management (SIEM) implementation». McGraw-Hill Osborne Media, 2010.

34. Miller D., Harris S., Harper A., VanDyke S., Blask C. «Security information and event management (SIEM) implementation». McGraw-Hill Osborne Media, 2010.

35. Sreemathy J., Joseph V.I., Nisha S., Prabha I.C., Priya R.M.G. «Data integration in ETL using Talend», 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1444–1448.

36. Mkhwanazi X., Le H., Blake E. «Clustering between data mules for better message delivery», 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 209–214.

37. Kumara I., Gamage C. «Towards reusing ESB services in different ESB architectures», 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, 2010, pp. 25–30.

38. Gnatyuk S., Berdibayev R., Aleksander M., Sydorenko V., Zhyharevych O., Polozhentsev A. «Software System for Cybersecurity Events Correlation and Incident Management in Critical Infrastructure». Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies. 2024. Vol. 213. P. 247-269. Springer, Cham.

39. Polozhentsev A., Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O. «Novel Cyber Incident Management System for 5G-based Critical Infrastructures». IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, P. 1037-1041.

РОЗДІЛ 4
ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНИХ МОДЕЛЕЙ ТА СИСТЕМИ КОРЕЛЮВАННЯ ПОДІЙ ТА УПРАВЛІННЯ ІТ-ІНЦИДЕНТАМИ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

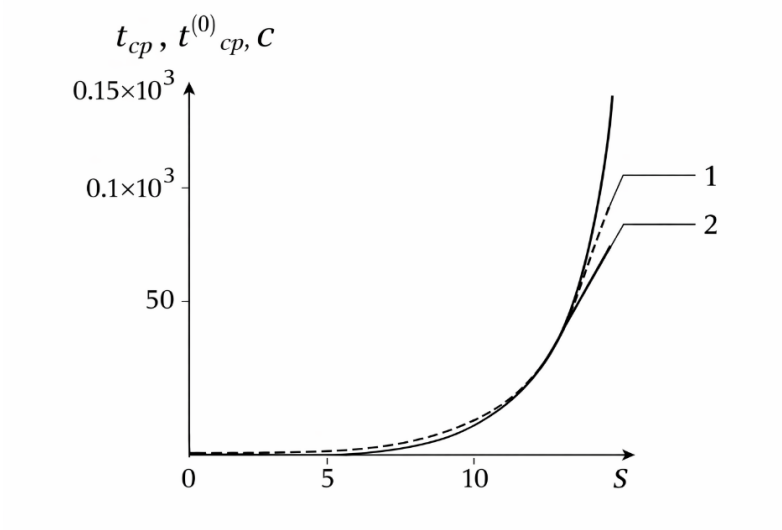
4.1. Експериментальне дослідження моделі оброблення даних у хмарних системах виявлення аномалій в критичній інфраструктурі

У роботі на основі результатів, наведених у [1-5], проведено експериментальне дослідження моделі оброблення даних у хмарних системах виявлення аномалій.

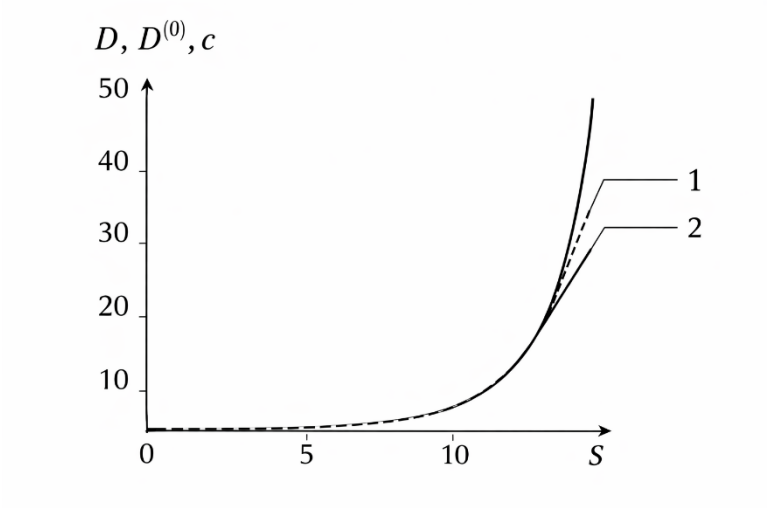
Розглянемо більш детально реалізацію запропонованої моделі.

Проведено дослідження взаємовпливу наведених у розділі 2.1, а також у виразах (2.2), (2.4), (2.6) та (2.7), тимчасових характеристик на загальний час оброблення метаданих та формування керуючих команд.

На рис. 4.1 представлені графіки загального часу $t_{ср}(s)$ (рис. 4.1 а) та часу обробки метаданих $t_{ср}^{(o)}(s)$ (рис. 4.1 б). А також графіки джиттера $D(s)$ загального часу (рис. 4.1 а) та часу обробки метаданих $D^{(o)}(s)$ (рис. 4.1 б) в умовах коли $a = 0,4; b = 0,7; h = 0,3;\ \ell = 1; p = 0,3; \lambda = 1200.$



а)



б)

Рис. 4.1 Графіки залежностей $t_{ср}(s)$ та $t_{ср}^{(o)}(s)$, $D(s)$ та $D^{(o)}(s)$

З рис. 4.1 випливає, що врахування часових характеристик формування керуючих сигналів дозволяє підвищити точність оцінювання часових параметрів до 1,7 раза, а показників джиттера — до 4,5 раза.

У більшості випадків щільність розподілу ймовірностей часу оброблення одного елемента метаданих та формування керуючої команди є одномодальною. Вирази (2.2) та (2.6) можуть бути використані для попередньої оцінки величини розкиду розподілу з урахуванням правила «трьох сигм». Водночас необхідність урахування зазначених факторів зумовлює застосування більш складних моделей.

Перспективним є використання графового підходу GERT-структур [1], що дозволяє оптимізувати структуру системи створення, передачі та оброблення метаданих, а також формування керуючих команд. Крім того, це забезпечує можливість оцінювання продуктивності системи та її масштабованості при збільшенні обсягу і складності розв’язуваних задач.

Експериментальне дослідження розробленої моделі та навчання нейронної мережі з точки зору виявлення аномалій

Вхідні та вихідні дані експерименту: вхідними даними є 20% набору даних NSL-KDD, вихідними — класифіковані дані (нормальні або аномальні, що відповідають загрозам).

Експериментальне середовище: середовище розробки з відкритим вихідним кодом для мови R, призначене для статистичного аналізу та візуалізації даних RStudio [5-9].

Середовище RStudio включає консоль, редактор коду з підтримкою підсвічування синтаксису та прямого виконання, а також інструменти для планування, налагодження та управління проєктами.

На рис. 4.2 наведено робоче середовище інструменту RStudio.

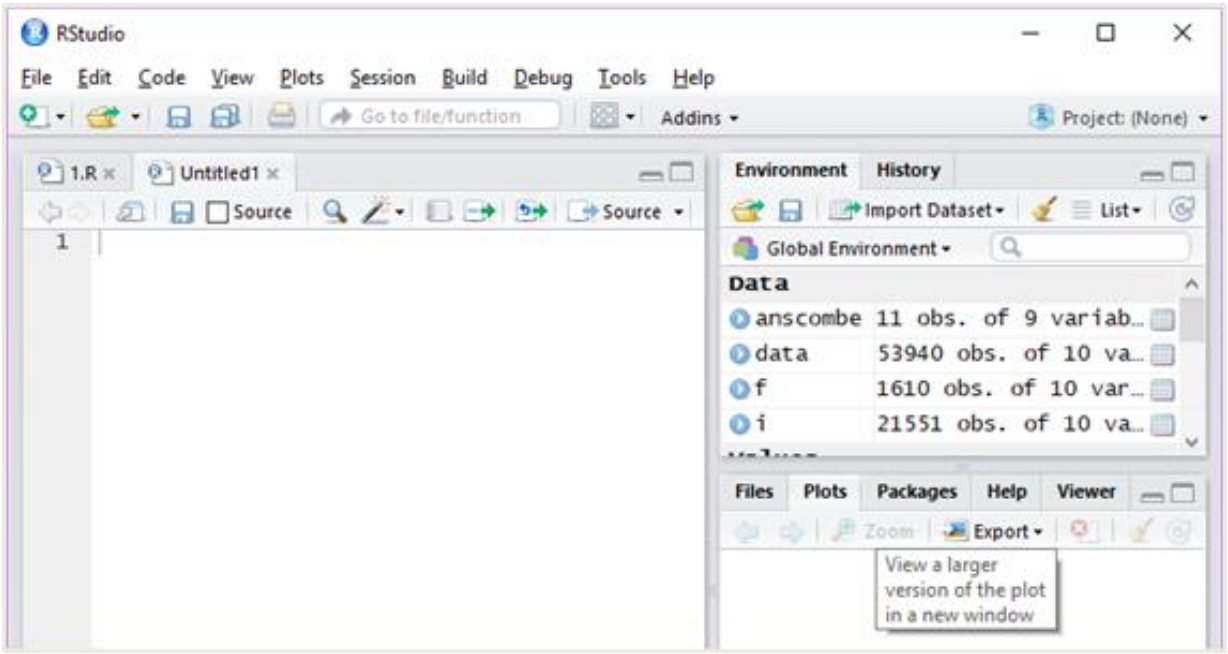


Рис. 4.2. Головне робоче вікно інструменту Rstudio

Вікно середовища поділено на чотири основні області:

1) робоча область для написання та запуску коду, що містить панель інструментів;

2) область перегляду даних, яка включає вкладки середовища (Environment) для відображення завантажених наборів даних і бібліотек, а також історії виконання команд;

3) консоль для відображення результатів виконання програмного коду та повідомлень середовища;

4) область відображення візуалізованих результатів (графіків, діаграм, гістограм тощо).

Етапи дослідження

Етап 1: Підключення всіх необхідних бібліотек, завантаження навчального набору бази даних NSL-KDD [7] (Рис. 4.3-Рис. 4.4).

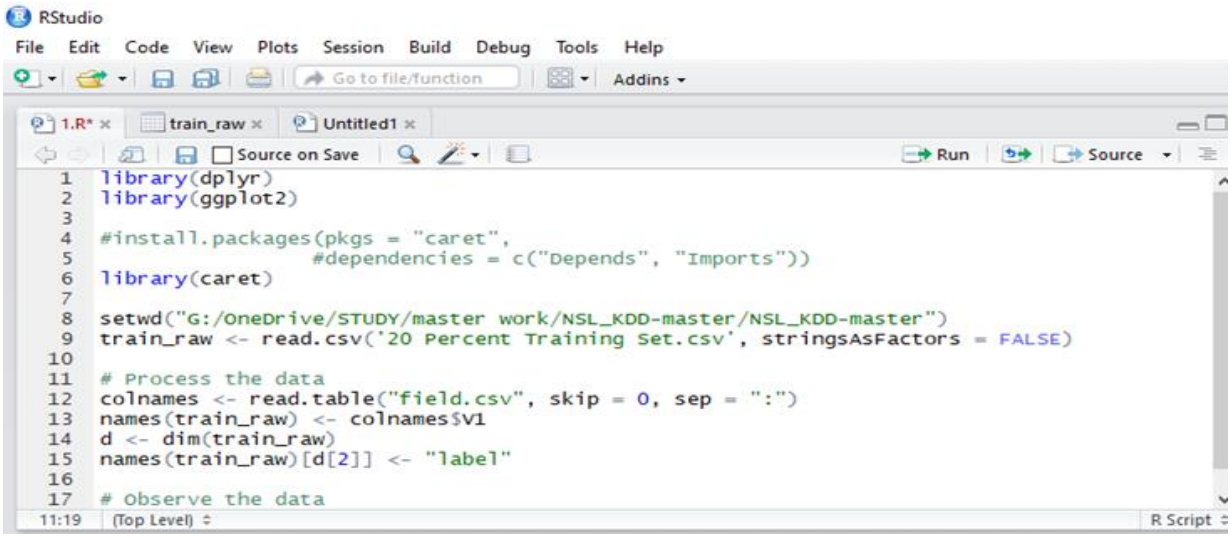
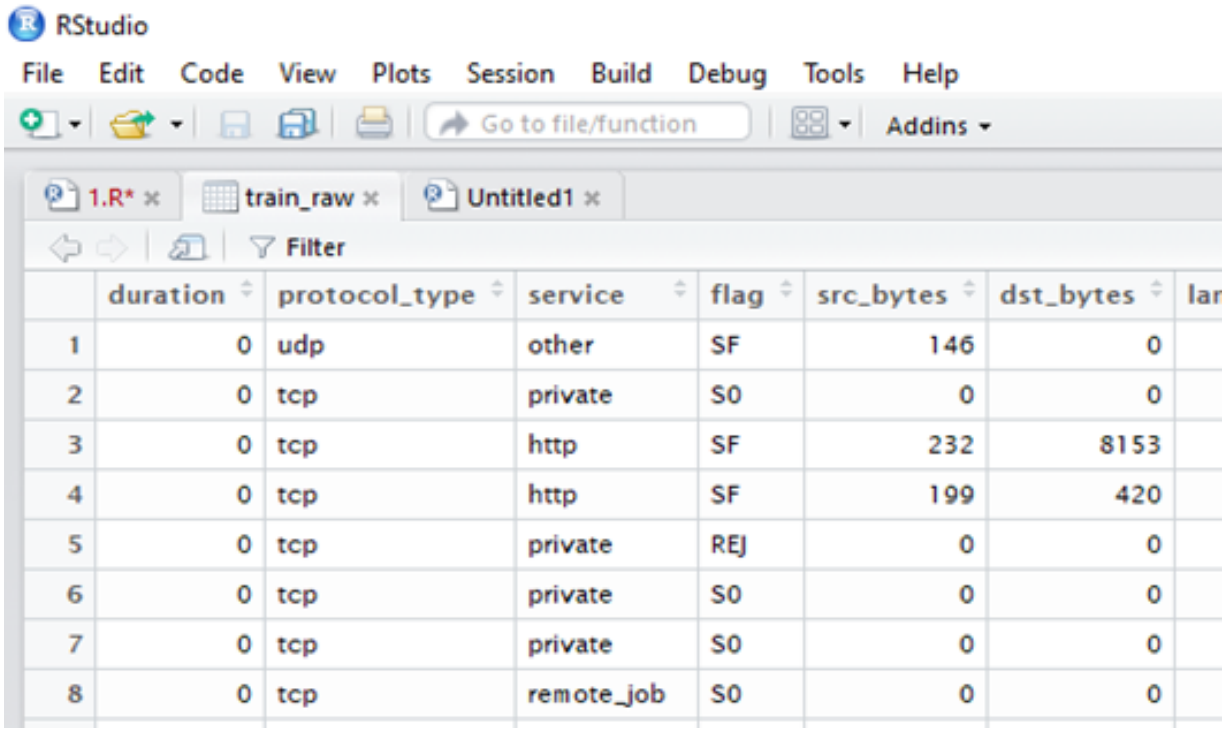


Рис. 4.3. Код для завантаження набору даних



	duration	protocol_type	service	flag	src_bytes	dst_bytes	lar
1	0	udp	other	SF	146	0	
2	0	tcp	private	S0	0	0	
3	0	tcp	http	SF	232	8153	
4	0	tcp	http	SF	199	420	
5	0	tcp	private	REJ	0	0	
6	0	tcp	private	S0	0	0	
7	0	tcp	private	S0	0	0	
8	0	tcp	remote_job	S0	0	0	

Рис. 4.4. Завантажено 20% навчальних даних

Етап 2: Аналіз набору тестових даних

Використано 20% навчальних даних бази даних NSL-KDD, що містять 25191 елемент, і мають 43 функції (Рис. 4.5).

```
Console G:/OneDrive/STUDY/master work/NSL_KDD-master/NSL_KDD-master/
> # Observe the data
> names(train_raw)
 [1] "duration"                   "protocol_type"
 [3] "service"                    "flag"
 [5] "src_bytes"                  "dst_bytes"
 [7] "land"                       "wrong_fragment"
 [9] "urgent"                     "hot"
[11] "num_failed_logins"          "logged_in"
[13] "num_compromised"            "root_shell"
[15] "su_attempted"               "num_root"
[17] "num_file_creations"         "num_shells"
[19] "num_access_files"           "num_outbound_cmds"
[21] "is_host_login"              "is_guest_login"
[23] "count"                      "srv_count"
[25] "serror_rate"                "srv_serror_rate"
[27] "rerror_rate"                "srv_rerror_rate"
[29] "same_srv_rate"              "diff_srv_rate"
[31] "srv_diff_host_rate"         "dst_host_count"
[33] "dst_host_srv_count"         "dst_host_same_srv_rate"
[35] "dst_host_diff_srv_rate"     "dst_host_same_src_port_rate"
[37] "dst_host_srv_diff_host_rate" "dst_host_serror_rate"
[39] "dst_host_srv_serror_rate"   "dst_host_rerror_rate"
[41] "dst_host_srv_rerror_rate"   NA
[43] "label"
```

Рис. 4.5. Атрибути навчального набору даних

На рисунку Рис. 4.6 показано розподіл типів загроз та атак.

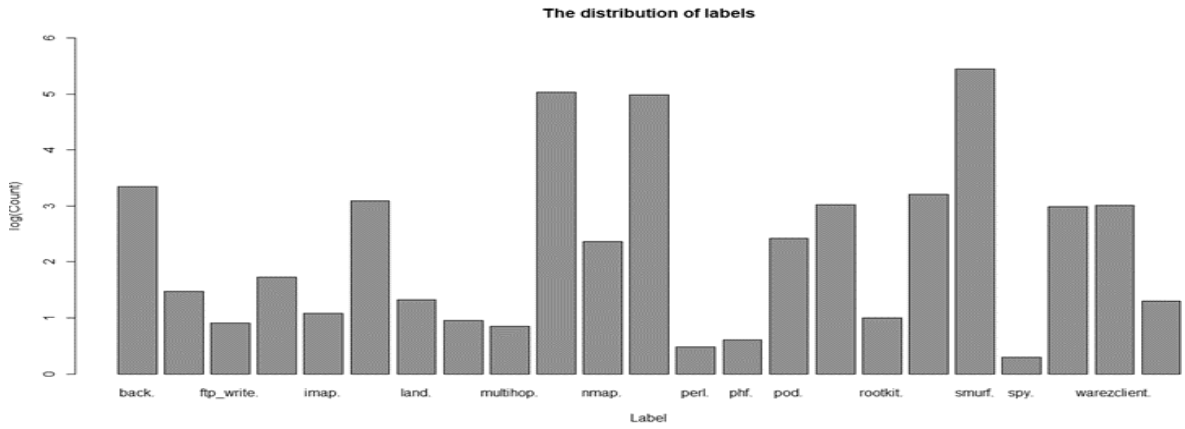


Рис. 4.6. Розподіл типів загроз

З рис. 4.6 випливає, що найбільша кількість атак належить до класу DoS.

Етап 3: Перевірка запропонованого методу

Для перевірки ефективності запропонованої моделі використано наявні функції оброблення даних та відповідний набір даних. На початковому етапі виконано попередню обробку даних, зокрема видалення дублікатів і залишкових записів, а також формування підвибірок даних (рис. 4.7).

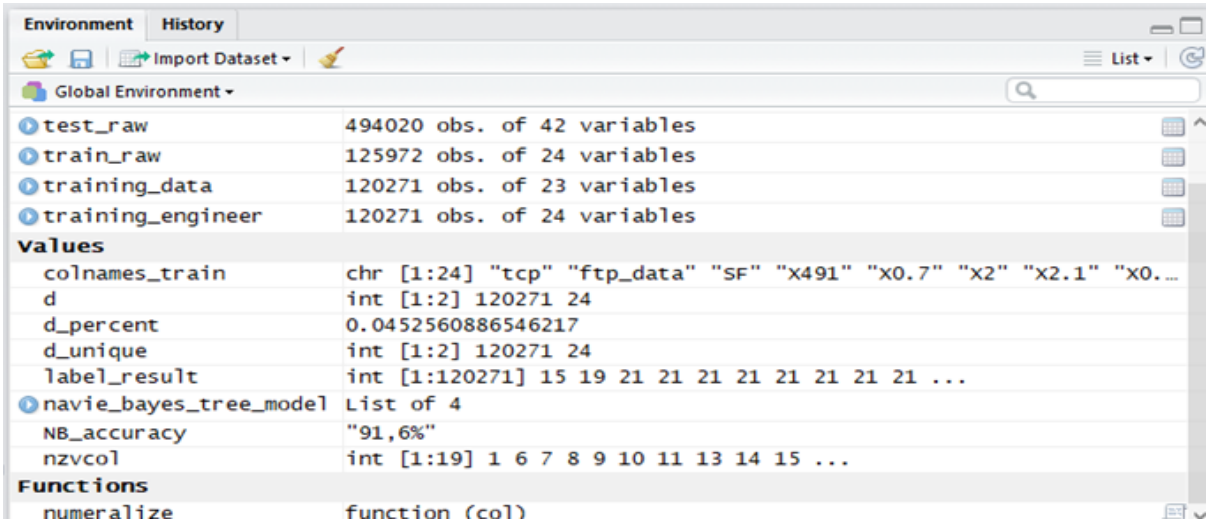


Рис. 4.7. Результати обробки даних

Після попередньої обробки здійснено аналіз мережевого трафіку (рис. 4.8), у результаті якого виконано розподіл даних на нормальні та аномальні.

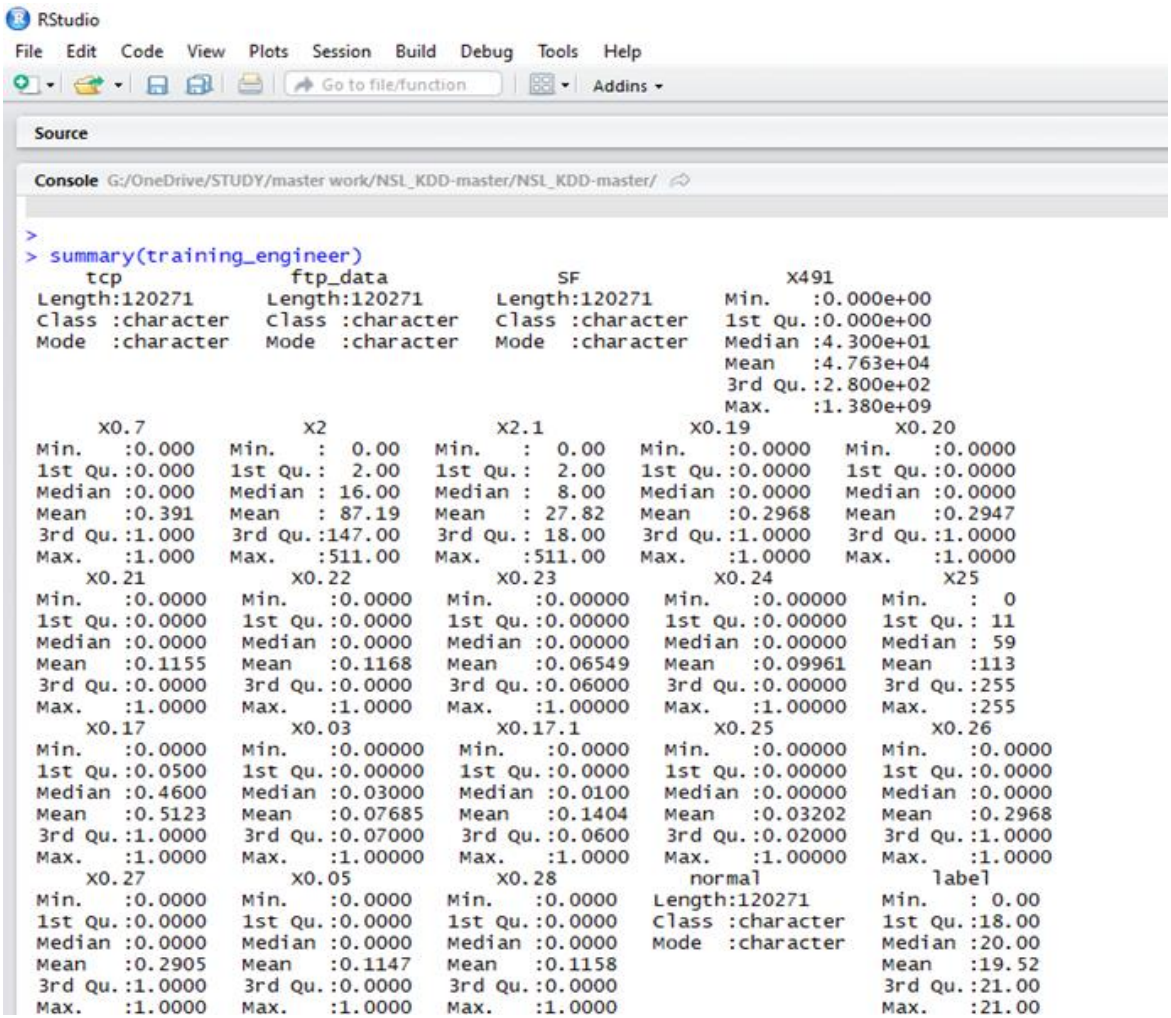


Рис. 4.8. Результат аналізу трафіка

Етап 4: Результат експерименту

У ході експериментального дослідження було отримано такі результати: загальний відсоток виявлених загроз – 96,36%, правильно класифікованих – 95,89%, неправильно класифікованих – 4,11% (рис. 4.9).

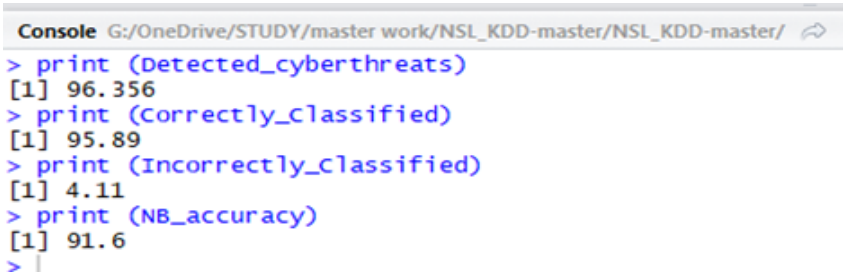


Рис. 4.9. Результати дослідження моделі

Оскільки у процесі дослідження певний відсоток загроз було класифіковано некоректно, доцільно використати апарат статистичних гіпотез, зокрема визначення помилок першого та другого роду [5-9].

Помилка першого роду полягає у відхиленні нульової гіпотези *H0*, коли вона є істинною. Помилка другого роду полягає у прийнятті нульової гіпотези *H0*, коли насправді істинною є альтернативна гіпотеза.

У даному контексті необхідно визначити, яка частка неправильно класифікованих об’єктів належить до реальних загроз, а яка – до нормального трафіку. Нехай базове значення частки помилок становить *IC0 = 5%*. Припустимо, що в результаті вдосконалення алгоритму частка помилок зменшується до *IC = 4,11%* при використанні більшого обсягу даних та проведенні 10 експериментів. При цьому дисперсія результатів становить *1%.* Для перевірки статистичної значущості отриманих результатів формулюється нульова гіпотеза *H0: IC = 5%*, яка перевіряється за відповідним статистичним критерієм (рис. 4.10).

Результат: ймовірність отримання помилок 1 типу – 5%, 2 типу – 12%.

Експериментальне дослідження у системі моделювання CloudSim

Вхідні та вихідні дані експерименту: вхідними даними є набір даних NSL-KDD та зафіксований мережевий трафік, вихідними — класифіковані

дані (нормальні або аномальні, що відповідають загрозам), а також показники ефективності моделі.

```
library(dplyr)
library(ggplot2)

n = 10                  # sample size
sigma = 1           # population standard deviation
sem = sigma/sqrt(n); sem    # standard error


alpha = .05             # significance level
mu0 = 5             # hypothetical lower bound
q = qnorm(alpha, mean=mu0, sd=sem);


mu =4.11            # assumed actual mean
pnorm(q, mean=mu, sd=sem, lower.tail=FALSE)
```

Рис. 4.10. Обчислення помилок 2-го типу

Експериментальне середовище: система моделювання CloudSim.

Платформа CloudSim є узагальненим та масштабованим інструментом моделювання, який забезпечує імітацію систем і інфраструктури хмарних обчислень, зокрема центрів обробки даних (ЦОД). Вона є розширенням базової функціональності платформи GridSim і надає можливість моделювання сховищ даних, веб-сервісів та розподілу ресурсів між віртуальними машинами [6]. На рис. 4.11 наведено журнал виявлених загроз, сформований під час проведення моделювання.

```
0,tcp,smtp,SF,1022,387,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,3,0,0,0,0,1,0,1,
255,28,0.11,0.72,0,0,0,0,0.72,0.04,normal,21
0,tcp,telnet,SF,129,174,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0
,255,255,1,0,0,0,0.01,0.01,0.02,0.02,guess_passwd,15
0,tcp,http,SF,327,467,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,33,47,0,0,0,0,1,0,0
.04,151,255,1,0,0.01,0.03,0,0,0,0,normal,21
0,tcp,ftp,SF,26,157,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,0,0,52,
26,0.5,0.08,0.02,0,0,0,0,0,guess_passwd,7
0,tcp,telnet,SF,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,255
,128,0.5,0.01,0,0,0,0,0.66,0.32,mscan,9
0,tcp,smtp,SF,616,330,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,1,0,1,2
55,129,0.51,0.03,0,0,0,0,0.33,0,normal,18
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,111,2,0,0,1,1,0.02,
0.07,0,255,2,0.01,0.07,0,0,0,0,1,1,neptune,21
0,tcp,telnet,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,120,120,1,1,0,0,1,0,0
,235,171,0.73,0.07,0,0,0.69,0.95,0.02,0,neptune,18
37,tcp,telnet,SF,773,364200,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1
,0,0,38,73,0.16,0.05,0.03,0.04,0,0.77,0,0.07,normal,14
0,tcp,http,SF,350,3610,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0,1,0,0,
71,255,1,0,0.01,0.04,0,0,0,0,normal,21
0,tcp,http,SF,213,659,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,24,24,0,0,0,0,1,0,0
,255,255,1,0,0,0,0,0,0,0,normal,21
0,tcp,http,SF,246,2090,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,16,16,0,0,0,0,1,0,
0,35,255,1,0,0.03,0.05,0,0,0,0,normal,21
0,udp,private,SF,45,44,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,505,505,0,0,0,0,1,
0,0,255,255,1,0,1,0,0,0,0,0,normal,15
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,204,18,0,0,1,1,0.09
,0.07,0,255,18,0.07,0.07,0,0,0,0,1,1,neptune,21
0,tcp,ldap,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,118,19,0,0,1,1,0.16,0.
```

Рис. 4.11. Журнал виявлених загроз

Аналіз журналів подій у середовищі RStudio проведено з метою візуалізації результатів, зокрема:

- розподілу ідентифікованих загроз та атак (рис. 4.12);
- відсоткового розподілу виявлених загроз залежно від їх типу (рис. 4.13).

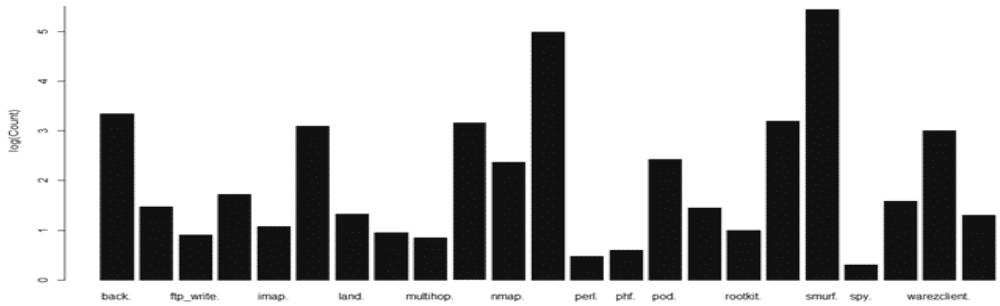


Рис. 4.12. Розподіл виявлених загроз та атак

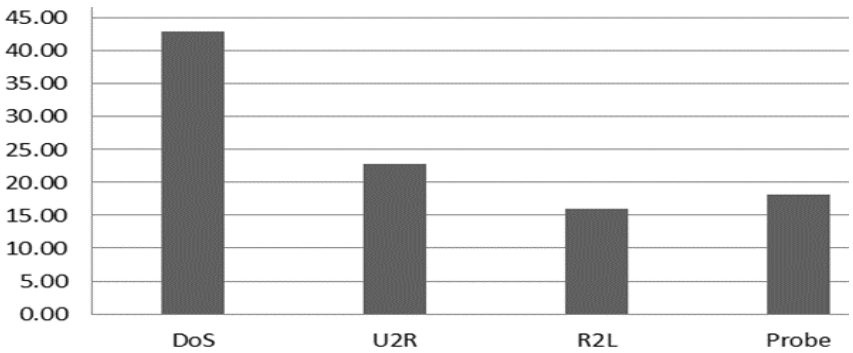


Рис. 4.13. Відсотковий розподіл виявлених загроз в залежності від їх типу

З Рис. 4.13 видно, що атаки, які найбільш часто відбуваються, відносяться до DoS.

Порівняння результатів моделювання на платформі CloudSim показано у таблиці 4.1

Таблиця 4.1

Порівняння результатів моделювання для модуля виявлення SIEM

Експеримент	**Використання моделі**	**Виявлені загрози**
1	–	45.87%
2	+	93.89%

Отримані результати свідчать, що при моделюванні ЦОД без використання розробленої моделі, навіть за наявності вбудованого механізму

виявлення загроз, рівень виявлення становить 45,87%, що є недостатнім. Натомість при використанні запропонованої моделі цей показник зростає до 93,89%, що підтверджує її ефективність [7–9].

Крім того на рис. 4.14 відображено порівняння результатів виявлення аномалій існуючих підходів та запропонаної моделі.

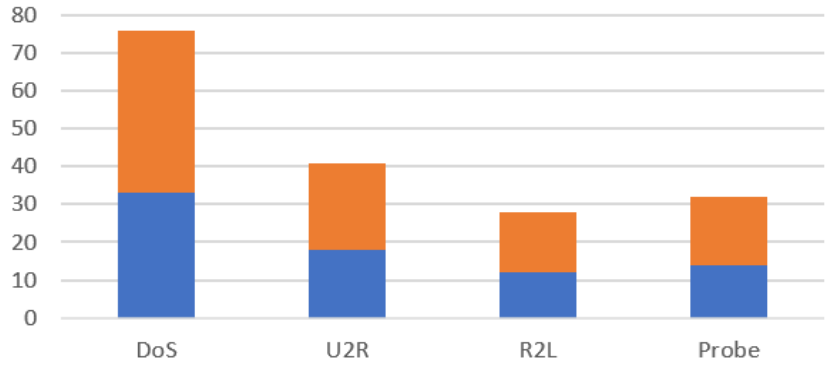


Рис. 4.14. Відсотковий розподіл виявлених загроз в залежності від їх типу

Таким чином, експериментально досліджено структурно-аналітичну модель оброблення даних, яка дозволяє оцінювати часові характеристики оброблення метаданих та формування керуючих команд із урахуванням передачі управління програмному клієнту ІКС, що забезпечує підвищення точності математичного моделювання. Ефективність моделі підтверджено експериментальними дослідженнями з використанням набору даних NSL-KDD: рівень виявлення загроз підвищено до 93,89% порівняно з 45,87% без застосування моделі, що відповідає приросту ефективності на 48,02%. Розроблені рішення можуть бути використані при побудові хмарних сервісів та SIEM-систем для ОКІ.

4.2. Експериментальне дослідження моделі онтологіко-реляційного сховища даних

У роботах [5, 10, 17] було проведено експериментальне дослідження моделі онтологіко-реляційного сховища даних.

Запропонована у роботі модель може бути реалізована у складі системи корелювання подій та управління IT-інцидентами [5]. При масштабуванні

ресурсів у розробленій системі корелювання подій та управління IT-інцидентами існує кілька практичних правил:

- Вузли системи корелювання подій та управління IT-інцидентами орієнтовані на потужність процесора. Вони також служать інтерфейсом користувача для браузера.
- Вузли Elasticsearch повинні мати якомога більше оперативної пам'яті та найшвидші диски, які можливо використати. Тут все залежить від швидкості введення-виведення.
- MongoDB зберігає метаінформацію та дані конфігурації і не потребує багато ресурсів.
- Отримані повідомлення зберігаються лише в Elasticsearch [10, 17].

Основним завданням онтологіко-реляційного сховища даних для роботи системи корелювання подій та управління IT-інцидентами є суміщення роботи двох типів баз даних та одночасне збереження можливості кластеризації баз даних обох типів.

Для обґрунтування вибору найбільш ефективних баз даних $\boldsymbol{DB}$, які використовуються в сучасних SIEM-системах, використаємо запропоновану модель онтологіко-реляційного сховища даних.

Відповідно до проаналізованих систем, при $n = 34$ з урахуванням (2.8), визначимо множину баз даних таким чином:

$$\mathbf{DB} = \{\bigcup_{i=1}^{34} DB_i\} = \{DB_1, DB_2, \ldots, DB_{34}\},$$

де DB_1 – Ariel database, DB_2 – PostgreSQL, DB_3 – SQLite, DB_4 – Oracle, DB_5 – SQL Server, DB_6 – MySQL, DB_7 – DB2/Linux, DB_8 – Informix, DB_9 – MemSQL, DB_{10} – AWS Aurora, DB_{11} – Microsoft SQL Server, DB_{12} – AWS RedShift, DB_{13} – SAP SQL Anywhere, DB_{14} – Sybase ASE, DB_{15} – Sybase IQ, DB_{16} – Teradata, DB_{17} – MSSQL, DB_{18} – Data Access Server (DAS), DB_{19} – DB2/UDB, DB_{20} – RedisDB, DB_{21} – Rap Sheet, DB_{22} – RabbitMQ, DB_{23} – MongoDB, DB_{24} –

ElasticSearch, DB_{25} – Kibana, DB_{26} – MS SQL Express, DB_{27} – MariaDB, DB_{28} – SQL, DB_{29} – DB2, DB_{30} – Own development CORR-E, DB_{31} – Microsoft SQL Azure, DB_{32} – SYBASE, DB_{33} – IBM, DB_{34} – Hadoop згідно [6, 17].

Відповідно, при $q = 7$ з урахуванням (2.9), визначимо множину запропонованих критеріїв:

$$\mathbf{EC} = \{\bigcup_{j=1}^{7} EC_q\} = \{EC_1,\ EC_2, EC_3, EC_4, EC_5, EC_6, EC_7\} =$$
$$= \{EC_{HOS},\ EC_{FL}, EC_{FS},\ EC_{STD}, EC_{SCD}, EC_{SSQL}, EC_{DBAAS}\} =$$
$$= \{HOS, FL, FS, STD, SCD, SSQL, DBAAS\},$$

де EC_1 – високоорганізована структура, EC_2 – гнучкість, EC_3 – швидкий доступ, EC_4 – підтримка різних типів даних, EC_5 – збереження даних конфігурацій, EC_6 – підтримка мови структурованих запитів, EC_7 – DBaaS (підтримка хмарних технологій).

Відповідно, при $p = 5$ з урахуванням (2.22), визначимо множину запропонованих критеріїв:

$$\mathbf{TS} = \{\bigcup_{k=1}^{5} TS_p\} = \{TS_1,\ TS_2, TS_{T3}, TS_4, TS_5\} =$$
$$= \{TS_{QPL},\ TS_{ES}, TS_{RSSI},\ TS_{OSF}, TS_{SBDA}\} = \{QPL, ES, RSSI, OSF, SBDA\},$$

де TS_1 – швидке опрацювання журналів, TS_2 – легкість масштабування, TS_3 – надійність зберігання службової інформації, TS_4 – оперативний пошук та фільтрація даних, TS_5 – здійснення комплексної бізнес-аналітики даних.

Для проведення процедури ранжування баз даних, відповідно до виразів (2.10-2.12) почергово будувались матриці попарних порівнянь.

Після чого відбулась побудова загального вектору критеріїв та оцінка ваги векторів відносно важливості кожного критерію (див. рис. 4.15) відповідно до (2.13-2.14).

	EC_1	EC_2	EC_3	EC_4	EC_5	EC_6	EC_7	Вектор	Вага
EC_1	1	1	0,33	1	0,33	1	1	0,731	0,098
EC_2	1	1	0,33	0,33	1	1	1	0,731	0,098
EC_3	3	3	1	1	1	1	1	1,369	0,184
EC_4	1	3	1	1	0,33	0,33	1	0,855	0,115
EC_5	3	3	1	3	1	0,33	1	1,369	0,184
EC_6	1	1	1	3	3	1	1	1,369	0,184
EC_7	1	1	1	1	1	1	1	1	0,135
	11	13	5,66	10,33	7,66	5,66	7	7,422	1,000

Рис. 4.15. Побудова вектору критеріїв та оцінка ваги векторів відносно важливості кожного критерію

Наступним кроком була побудова матриць вектору критеріїв та оцінка ваги векторів для кожного елемента множини баз даних DB та множини елементів критеріїв EC. А також, визначення рангу найбільш ефективних баз даних DB, результат яких відображено на рис. 4.16.

	EC_1	EC_2	EC_3	EC_4	EC_5	EC_6	EC_7	RN_{DB}
	0,098	0,098	0,184	0,115	0,184	0,184	0,135	
DB_1	0,54	0,2	0,3	0,12	0,5	0,6	0,37	0,39
DB_2	0,27	0,24	0,35	0,17	0,12	0,43	0,15	0,69
DB_4	0,49	0,37	0,2	0,54	0,34	0,32	0,14	0,68
DB_5	0,2	0,3	0,16	0,34	0,33	0,28	0,22	0,66
DB_6	0,5	0,21	0,39	0,37	0,44	0,46	0,12	0,65
DB_{23}	0,22	0,37	0,38	0,4	0,42	0,38	0,42	0,77
DB_{24}	0,31	0,23	0,2	0,23	0,32	0,31	0,48	0,78
DB_{31}	0,6	0,35	0,5	0,3	0,18	0,47	0,21	0,69

Рис. 4.16. Визначення рангу найбільш ефективних баз даних

В результаті експериментальних досліджень (див. рис. 4.16) були обгрунтовано виділені найбільш ефективні бази даних: DB_{23} – MongoDB та DB_{24} – ElasticSearch, що відповідають множині критеріїв та можуть вирішувати множину необхідних задач, представлених на рис. 4.17.

Основним завданням онтологіко-реляційного сховища даних у системі корелювання подій та управління IT-інцидентами на ОКІ є забезпечення інтеграції двох типів баз даних із збереженням можливості їх кластеризації та узгодженої роботи.

Рис. 4.17. Схема результату вибору найбільш ефективних БД

У ході експериментальних досліджень обґрунтовано використання двох типів баз даних:

1) Тип бази даних 1 – для швидкого оброблення журналів

Для реалізації цього завдання використано технологію Elasticsearch — відкриту платформу для повнотекстового пошуку та аналітики даних. Вона забезпечує індексацію та ефективний пошук, сортування і фільтрацію даних, що дозволяє реалізувати інший, порівняно з традиційними реляційними СУБД, підхід до оброблення інформації.

Дані в Elasticsearch зберігаються у вигляді документів у форматі JSON, що забезпечує зручну інтеграцію з різними мовами програмування. Архітектура системи базується на бібліотеці Lucene та орієнтована на оброблення великих обсягів неструктурованих даних. У межах дослідження виконано доопрацювання програмного коду мовою Java для адаптації системи до специфіки задачі.

2) Тип бази даних 2 – для зберігання службової інформації

Для зберігання метаданих і конфігураційної інформації використано технологію MongoDB – документоорієнтовану NoSQL СУБД, яка не потребує жорстко визначеної схеми даних та використовує JSON-подібні документи. MongoDB підтримує механізми реплікації, що забезпечують відмовостійкість системи: у разі відмови основного вузла автоматично обирається новий головний вузол. Крім того, система підтримує горизонтальне масштабування шляхом сегментування даних між вузлами кластера, що дозволяє ефективно розподіляти навантаження.

Додатково MongoDB може використовуватися як файлове сховище з підтримкою розподілу даних і реплікації (GridFS), що забезпечує роботу з великими обсягами файлів [10, 17].

Таким чином, розроблена модель онтологіко-реляційного сховища даних забезпечує підвищення ефективності зберігання, класифікації та оброблення даних, а також високий рівень продуктивності при роботі з великими обсягами інформації.

Методика зберігання та класифікації даних

На основі запропонованої моделі онтологіко-реляційного сховища даних створено методику зберігання та класифікації даних (рис. 4.18), яка дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірностей та забезпечувати високу швидкість пошуку.

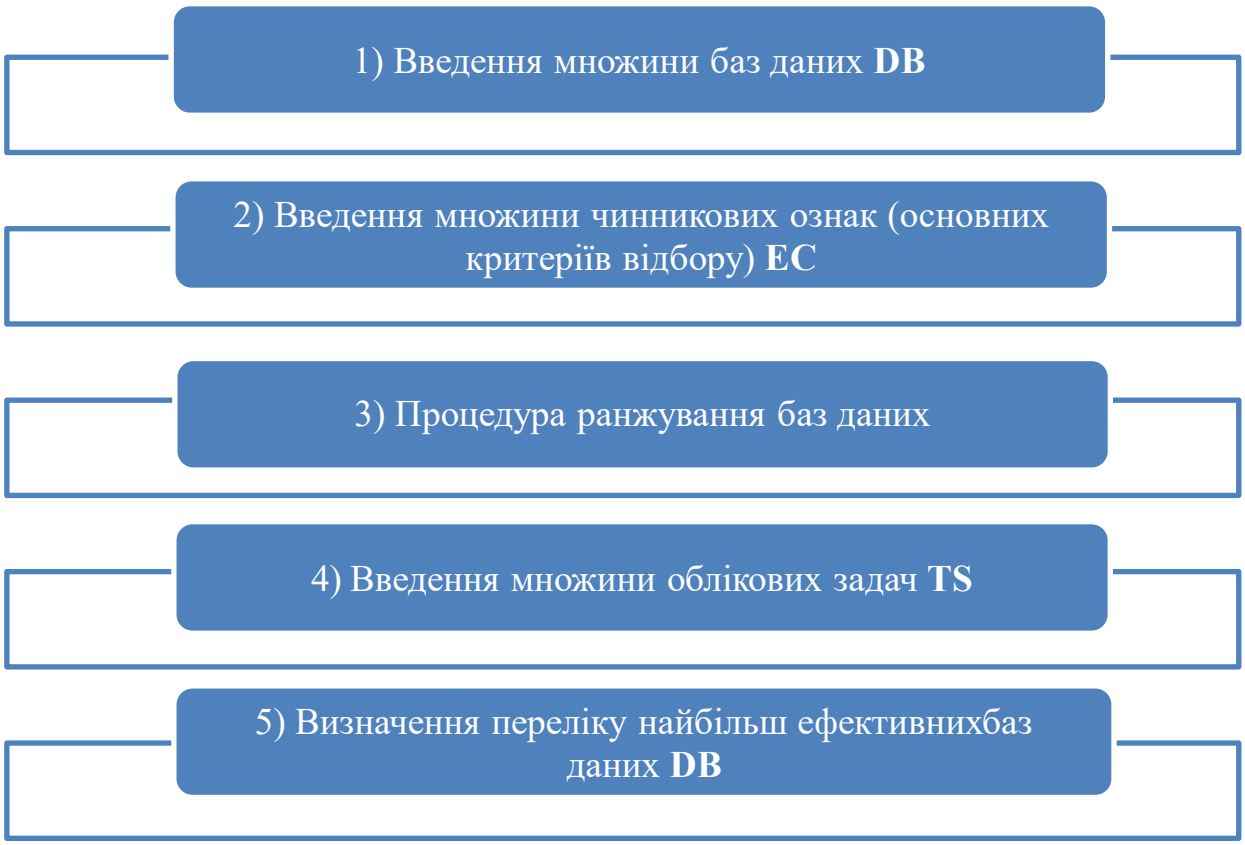


Рис. 4.18. Методика зберігання та класифікації даних

Методика складається з наступних кроків: 1) Введення множини баз даних **DB**. 2) Введення множини чинникових ознак (основних критеріїв відбору) **EC**. 3) Процедура ранжування баз даних. 4) Введення множини облікових задач **TS**. 5) Визначення переліку найбільш ефективних БД, які використовуються в сучасних SIEM-системах.

Отже, за допомогою наведеної розрахункової процедури було досліджено множину баз даних **DB**, які використовуються в сучасних SIEM-системах, за основними критеріями **EC**.

В результаті чого були досліджені 34 **DB** за 7 критеріями **EC**, та обгрунтовано виділені дві найбільш ефективні **DB** – MongoDB та ElasticSearch, які дозволяють сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірностей та забезпечувати високу швидкість пошуку.

4.3. Експериментальне дослідження моделі інтеграційної шини даних

Приклад впровадження розробленої моделі інтеграційної шини даних представлено в роботах [5, 11].

Реалізовано та впроваджено модель ІШД для ефективного функціонування SIEM-систем на ОКІ. Розроблено ІШД на базі сервіс-орієнтованої архітектури. Платформа використовує розподілені бази даних різних типів для розв'язання паралельних завдань із контролю метрик і подій. Це на порядок збільшує параметри, забезпечуючи: швидкість опрацювання великих потоків інформації; мінімальні затримки на опрацювання даних; мінімальні затримки для побудови аналітичних звітів і запитів; високу відмовостійкість; розширюваність сховища шляхом простого додавання вузлів без простою бази. Використання API значно спрощує взаємодію, об’єднуючи можливості різних сервісів, та утворюючи доступні різним користувачам інтерфейси.

Відповідно до проаналізованих сервісів систем, при $t = 8$ з урахуванням (3.1), було визначимо множину сервісів таким чином:

$\mathbf{SR} = \{\bigcup_{s=1}^{8} SR_s\} = \{SR_1, SR_2, \ldots, SR_8\}$, де SR_1 – збір та зберігання подій, які надходять до системи, SR_2 – виявлення та розбір інцидентів безпеки, SR_3 – виявлення атак та порушень політики безпеки, SR_4 – оцінка захищеності ресурсів системи, що контролюються (у т.ч. КВР), SR_5 – пошук та управління вразливостями, SR_6 – формування звітів, SR_7 – підтримка роботи з хмарними середовищами, SR_8 – розширені можливості пошуку.

Розрахований за (3.3) ранг критичності сервісів SR для визначених найбільш ефективних баз даних: DB_{23} – MongoDB та DB_{24} – ElasticSearch представлено у табл. 4.2.

Таблиця 4.2

Значення рангу критичності сервісів SR для двох баз даних DB_{23} та DB_{24}

	SR_1	SR_2	SR_3	SR_4	SR_5	SR_6	SR_7	SR_8
RN ElasticSearch	125	95	85	70	57	120	145	125
RN MongoDB	95	90	75	58	75	127	140	126

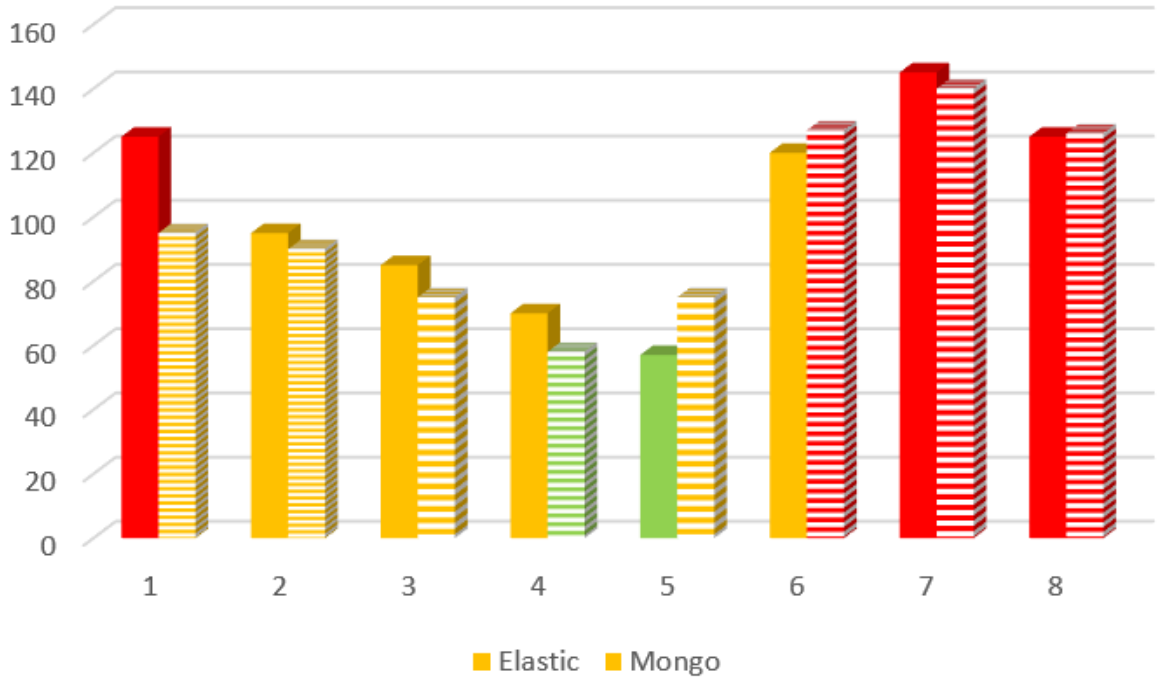
Процес ранжування критичних сервісів SIEM-системи за допомогою діаграми Парето наведено на рис. 4.19: а) представлення критичності сервісів SIEM-системи у порядку їх появи сервісів; б) ранжований перелік сервісів за упорядкований за рівнем критичності.

Розроблено модель ІШД на базі сервіс-орієнтованої архітектури. Платформа використовує розподілені бази даних різних типів для забезпечення паралельного виконання завдань контролю метрик і оброблення подій.

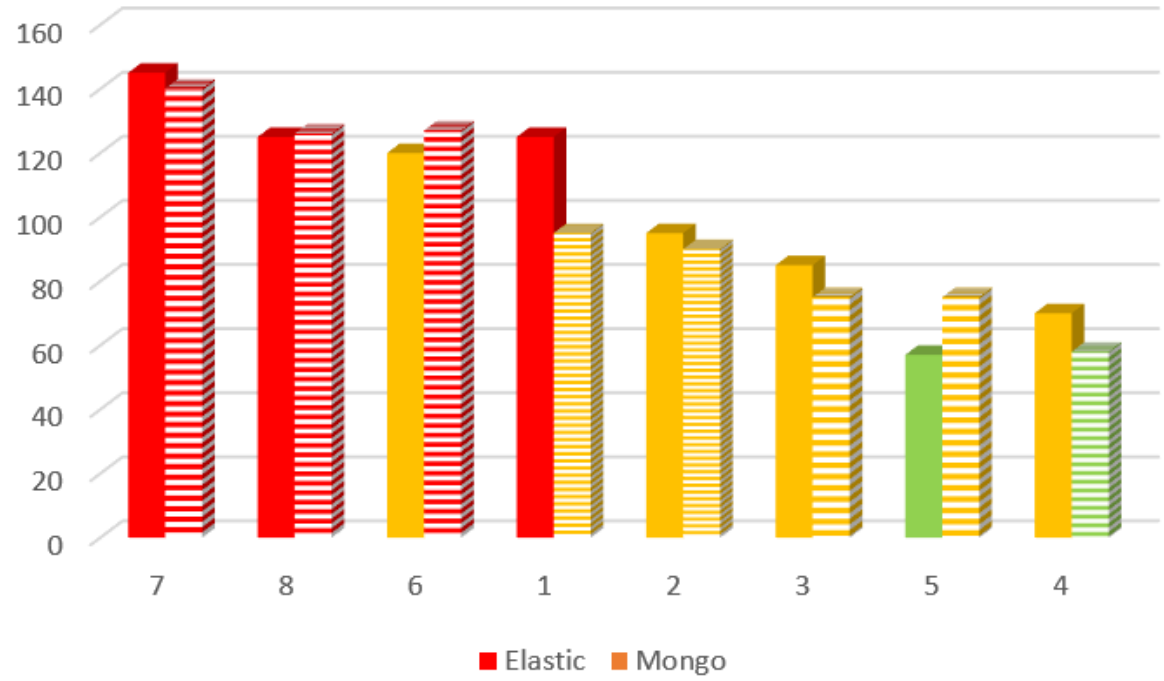
Застосування запропонованого підходу забезпечує:

- підвищення швидкості оброблення великих потоків даних;
- зменшення затримок під час оброблення інформації;
- скорочення часу формування аналітичних звітів і виконання запитів;
- підвищення відмовостійкості системи;

- масштабованість сховища даних шляхом додавання вузлів без зупинки системи.



а)



б)

Рис. 4.19. Процес ранжування критичних сервісів SIEM-системи

Використання API забезпечує ефективну взаємодію між компонентами системи, інтеграцію різних сервісів та формування доступних

користувацьких інтерфейсів. Отже, застосування розробленої ІШД для функціонування SIEM-систем забезпечує: гнучку маршрутизацію даних; гарантовану доставку повідомлень; організацію захищених каналів передавання інформації; централізоване управління; можливість моніторингу та діагностики процесів передачі даних; інтеграцію зі сторонніми системами обміну повідомленнями [11].

Специфікація SIEM-систем

З урахуванням [12-16] специфікація SIEM систем може бути відображена у вигляді основних та додаткових вимог:

1. Основні вимоги:

1.1. Системи спеціального призначення.

SIEM призначена для моніторингу та аналізу подій ІБ і повинна:

- здійснювати централізований збір, зберігання та обробку подій системних журналів (логів), а також мережевих потоків з різних систем інфраструктури Замовника;
- виділяти в загальному масиві даних важливі події та інциденти ІБ, що повинно дозволити фахівцям з ІБ Замовника, сконцентруватися на найбільш серйозних інцидентах і своєчасно реагувати на них;
- інформувати персонал Замовника про виявлені інциденти інформаційної безпеки шляхом надсилання повідомлень на електронну пошту.

1.2. Системи з централізованим управлінням.

SIEM повинна забезпечувати централізоване управління всіма своїми компонентами і функціоналом через єдиний графічний веб-інтерфейс.

1.3. Візуалізація даних (Dashboards). SIEM:

- дозволяє створювати графічні панелі (дашборди) за будь-якими подіями, з автоматичним оновленням із заданим інтервалом;
- підтримує створення нових графічних панелей або модифікацію існуючих за допомогою "майстра", методом, що не вимагає використання мов програмування;

- дозволяє зберігати графічні панелі для колективного використання. Графічні панелі повинні підтримувати різні типи представлення даних: таблиці, кругові та лінійні діаграми тощо, вони повинні функціонувати автоматично, без необхідності регулярного обслуговування оператором;
- підтримка відображення графічних панелей через WEB;
- підтримка інтерфейсу.

1.4. *Підтримка API*:

повинна мати відкритий програмний інтерфейс API для можливості інтеграції з іншими модулями.

1.5. Підтримка автентифікації та авторизації.

SIEM повинна підтримувати наступні методи для забезпечення автентифікації та авторизації користувачів:

- токени (для доступу до API);
- локальна база користувачів;
- Active Directory;
- LDAP.

1.6. Підтримка оновлень.

SIEM повинна підтримувати можливість автоматичного та/або ручного оновлення по мірі виходу нових версій.

1.7. Відмовостійкість.

База даних SIEM повинна підтримувати кластерну організацію в кількості не менше двох вузлів (node).

1.8. Масштабування.

SIEM:

- забезпечує горизонтальне масштабування шляхом додавання обладнання та, за необхідності, придбання додаткових ліцензій на SIEM відповідно до чинної (на момент масштабування) політики ліцензування;
- має компонент зберігання подій (базу даних), в якому реалізовані наступні функції:

- Масштабування без фіксованого ліміту на обсяг зберігання подій (додавання додаткового обладнання при необхідності).
- Відмовостійка реалізація.

1.9. Збір та фільтрація подій.

SIEM:

- підтримує стандартні методи збору логів подій: Syslog, Raw/Plaintext, GELF, CEF, файлові журнали подій (за допомогою агентів для Linux/Windows);
- підтримує аналіз подій у реальному часі;
- забезпечує фільтрацію, а також відображення через користувацький інтерфейс події в реальному часі, де користувач може одразу застосувати фільтри;
- зберігає критерії пошуку для швидкого доступу;
- підтримує пошук за подіями з використанням мови запитів (якщо ви використовуєте власну мову запитів, вона має бути описана в документації);
- надає користувачеві можливість самостійно підключати джерела подій, які не підтримуються за замовчуванням;
- підтримує передачу даних від джерел до системи керування захищеним каналом (якщо в протоколі є підтримка захищеної передачі);
- підтримує централізоване управління агентами через інтерфейс SIEM (для агентів сімейства Beats).

1.10. Вимоги до управління акаунтом.

SIEM:

- підтримує рольову модель управління із заздалегідь визначеним набором ролей;
- має можливість створювати та використовувати групи користувачів (Teams);
- має систему управління токенами для авторизації в API.

1.11. Вимоги до кількості подій в секунду (EPS):

- Середньоденний - не більше 200 EPS;
- Максимальний у найбільш завантажену годину - не більше 400 EPS.

1.12. Вимоги до інформації, що зберігається в базі.

Щоденний обсяг інформації, що зберігається в базі даних подій, становить не більше 4 ГБ на добу; Термін зберігання подій у базі даних та/або архіві SIEM - не менше 3 років.

2. Додаткові вимоги:

2.1. Постачальник забезпечує встановлення програмного забезпечення SIEM на фізичних серверах та/або платформі віртуалізації Замовника.

2.2. Постачальник надає Замовнику розрахунок потреб у ресурсах для встановлення SIEM. Розрахунок здійснюється Постачальником на основі вимог до продуктивності, зазначених у цій специфікації

2.3. Замовник забезпечує виділення ресурсів відповідно до зазначеного розрахунку, встановлення та базове налаштування операційних систем (включаючи налаштування дискової підсистеми та мережевих інтерфейсів) для встановлення SIEM. Постачальник надає Замовнику дистрибутив операційної системи (на носії або у вигляді посилання для завантаження) та, за необхідності, ліцензію на операційні системи.

2.4. Замовник забезпечує доступ з серверів до мережі Інтернет на час встановлення та налаштування SIEM.

2.5. На весь період експлуатації SIEM Замовник забезпечує постійний доступ з серверів SIEM до сервера ліцензування для контролю ліцензії.

2.6. Постачальник протягом десяти календарних днів завершує встановлення та налаштування системи SIEM.

4.4. Експериментальне дослідження системи корелювання подій та управління ІТ-інцидентами на ОКІ.

Розроблена у [5-6, 18-19] система корелювання подій та управління ІТ-інцидентами автоматизує визначення пріоритетів загроз безпеці та порушень вимоги ІБ на основі аналізу та кореляції подій ІБ. Система аналізує кожен вхід у систему та вихід із неї, доступ до ресурсів (у т.ч. КВР), запити до бази та транзакції тощо.

Система забезпечує: збирання, зберігання та аналіз подій з будь-якого джерела та у необхідний час; аналізу подій та виявлення незвичайних чи несанкціонованих дій; графічні панелі моніторингу подій, що настроюються; API для інтеграції зі сторонніми системами та сервісами.

4.4.1. Архітектура системи

Система має високу гнучкість і горизонтальну масштабованість. Для зберігання подій використовується NoSQL СУБД Elasticsearch, для зберігання всієї інформації про конфігурацію та правила використовується NoSQL СУБД mongoDB. Система, залежно від вимог до продуктивності та надійності, може бути розгорнута у різних варіантах. Приклади архітектури системи корелювання подій та управління IT-інцидентами наведено на рис. 4.20 та рис. 4.21 [5-6].

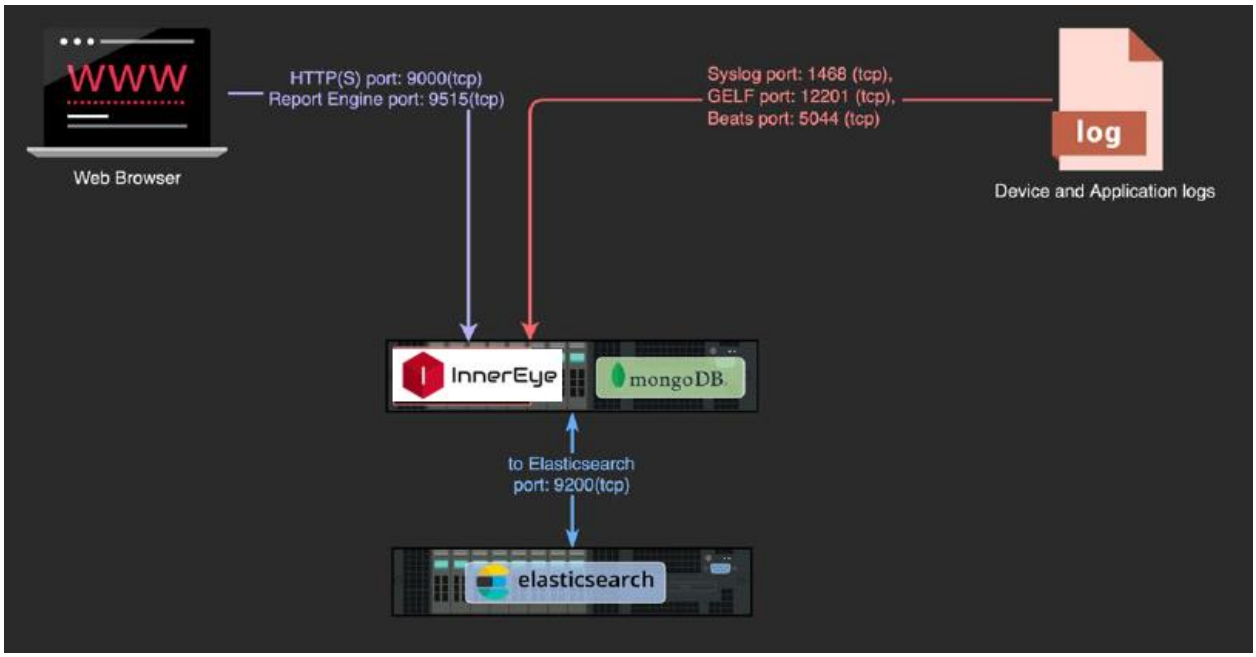


Рис.4.20. Приклад архітектури системи корелювання подій та управління IT-інцидентами з продуктивністю до 310 Гб подій на добу

Використання кластерів у системі корелювання подій та управління IT-інцидентами забезпечує високу продуктивність та надійність системи в цілому та дозволяє гнучко адаптувати систему до конкретних умов.

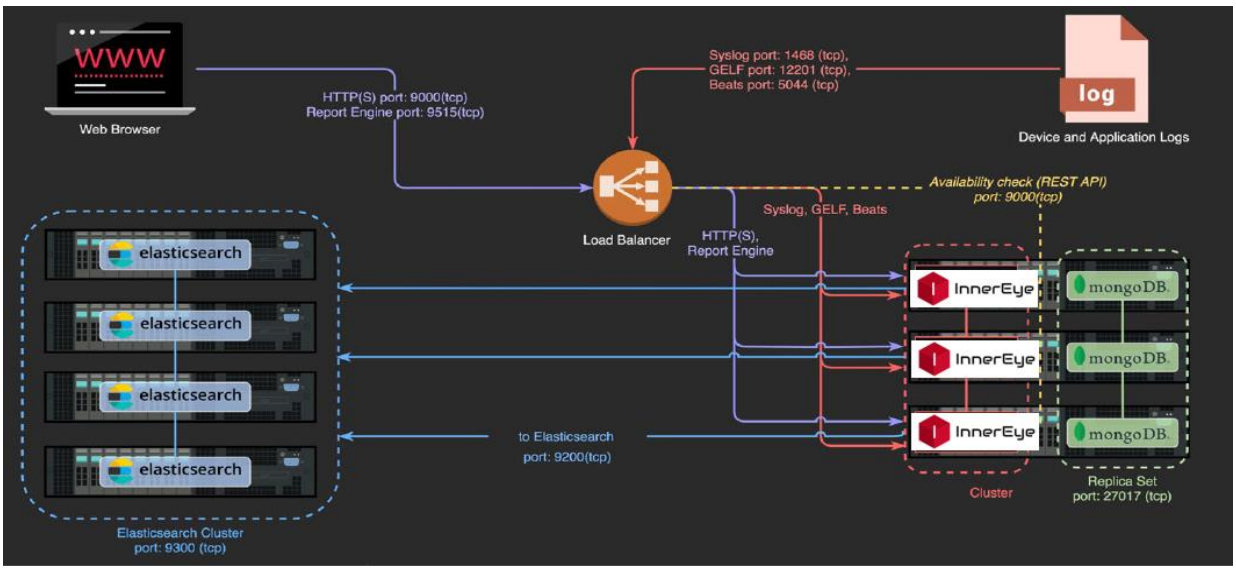


Рис.4.21. Приклад архітектури системи корелювання подій та управління IT-інцидентами з продуктивністю до 300 Гб подій на добу

4.4.2. Функціональні можливості системи

Джерела подій та робота з ними

Система підтримує стандартні методи збирання журналів подій (рис. 4.22).

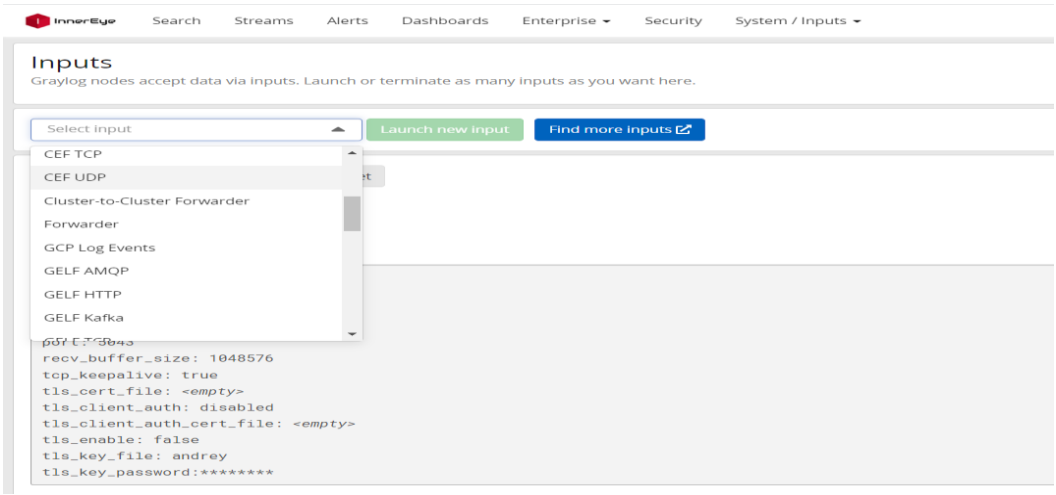


Рис.4.22. Меню вибору типу джерела подій

Журнали подій системи складаються з таких компонентів:

- Syslog (TCP, UDP, AMQP, Kafka);
- GELF (TCP, UDP, AMQP, Kafka, HTTP);
- AWS (AWS Logs, FlowLogs, CloudTrail);

- Beats/Logstash;
- CEF (TCP, UDP, AMQP, Kafka);
- JSON Path from HTTP API;
- Netflow/IPFIX (UDP);
- Plain/Raw Text (TCP, UDP, AMQP, Kafka).

До кожного типу джерела подій можливе створення індивідуального приймача (далі – Input) з індивідуальними параметрами (Рис. 4.23).

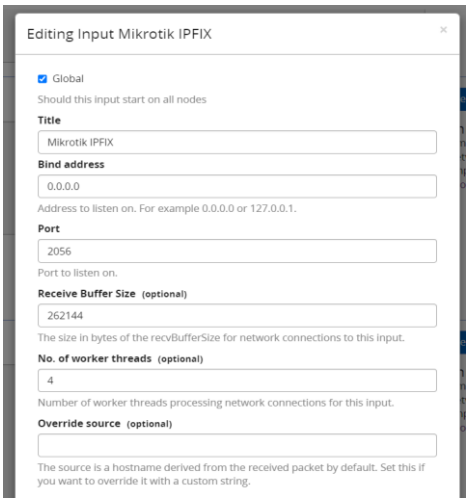


Рис. 4.23. Налаштування параметрів джерела подій

За кожним джерелом подій можна переглянути загальну інформацію, а також перейти до перегляду подій від цього джерела [5-6].

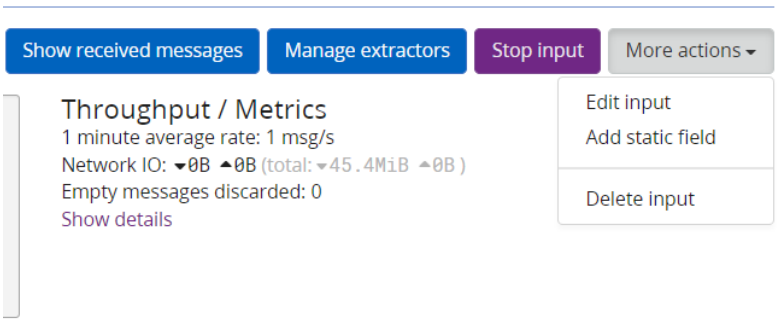


Рис. 4.24. Інформація про джерело подій та можливі дії з ним

Крім того, для протоколів, що підтримують безпечне передавання даних (TLS), можливе налаштування відповідних параметрів.

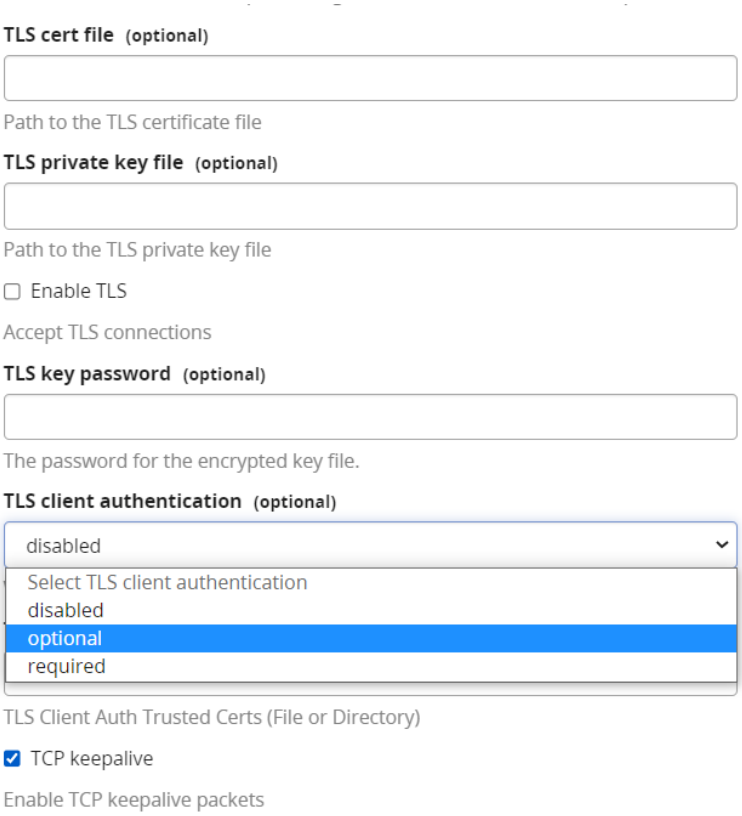


Рис. 4.25. Налаштування параметрів TLS

Збереження подій у базі даних

Система корелювання подій та управління IT-інцидентами зберігає події в NoSQL СУБД Elasticsearch. Можливе створення довільної (в рамках обмежень самої СУБД Elasticsearch) кількості баз даних (індексів) (Рис. 4.26).

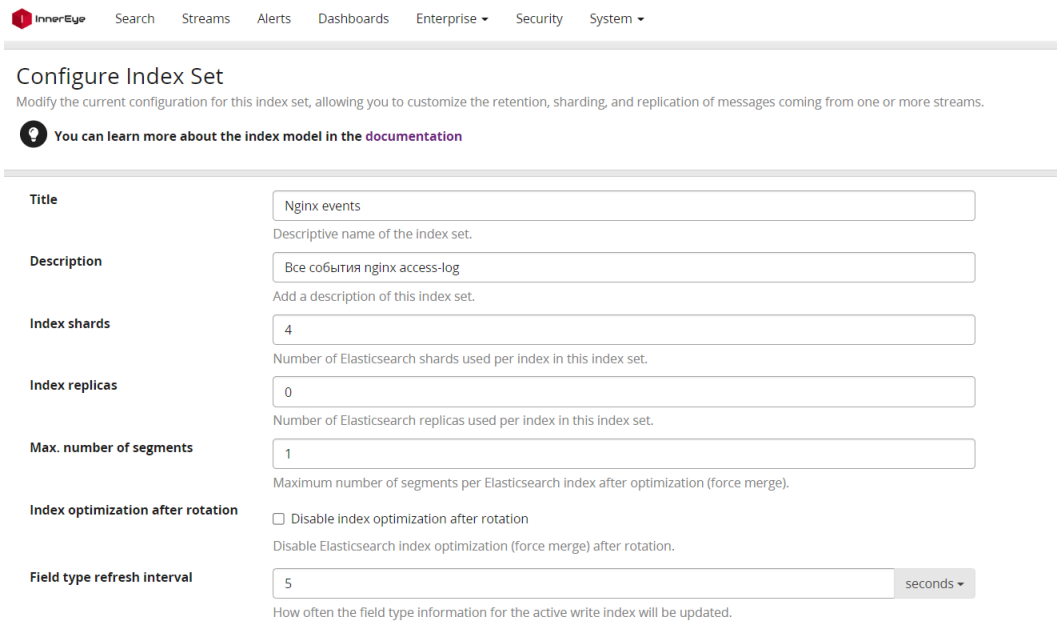


Рис. 4.26. Налаштування параметрів бази даних (індексу)

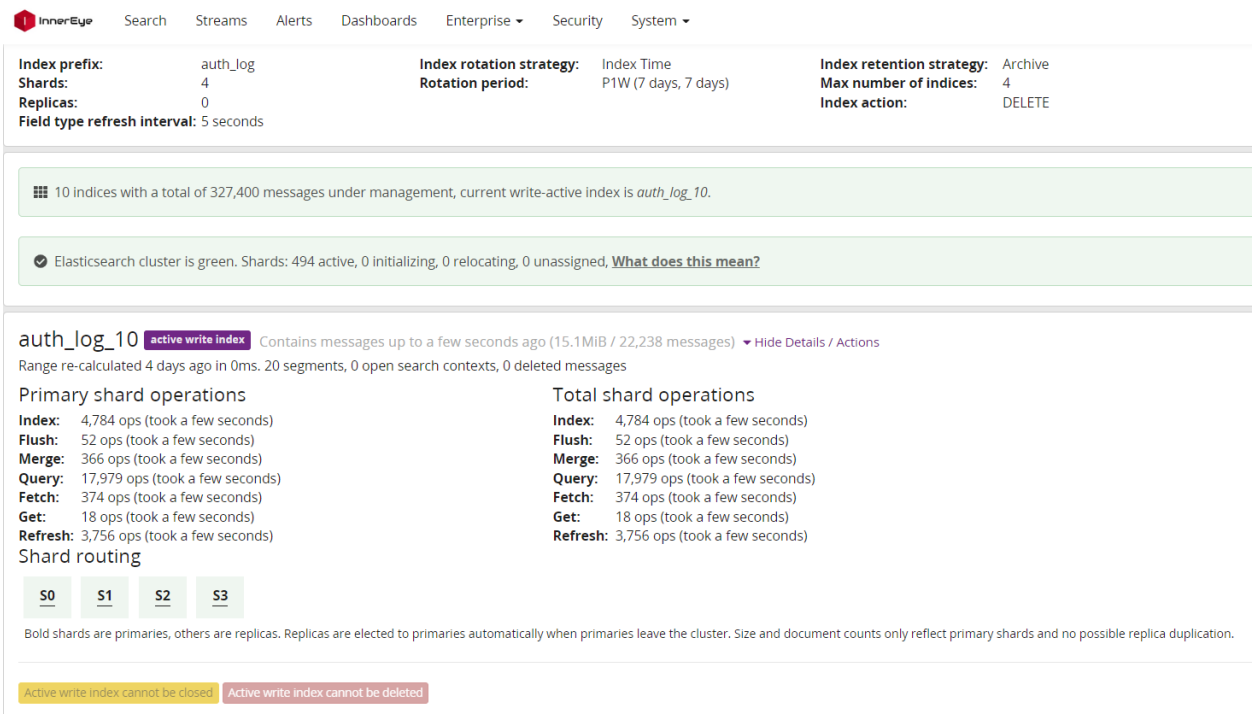


Рис. 4.27. Інформація щодо поточного стану бази даних (індексу).

Обробка подій – потоки

Потоки – це механізм, який розподіляє події за категоріями реального часу під час їх обробки (Рис. 4.28). Можливе створення довільної (в рамках фізичних обмежень серверного обладнання та обмежень СУБД mongoDB) кількості потоків.

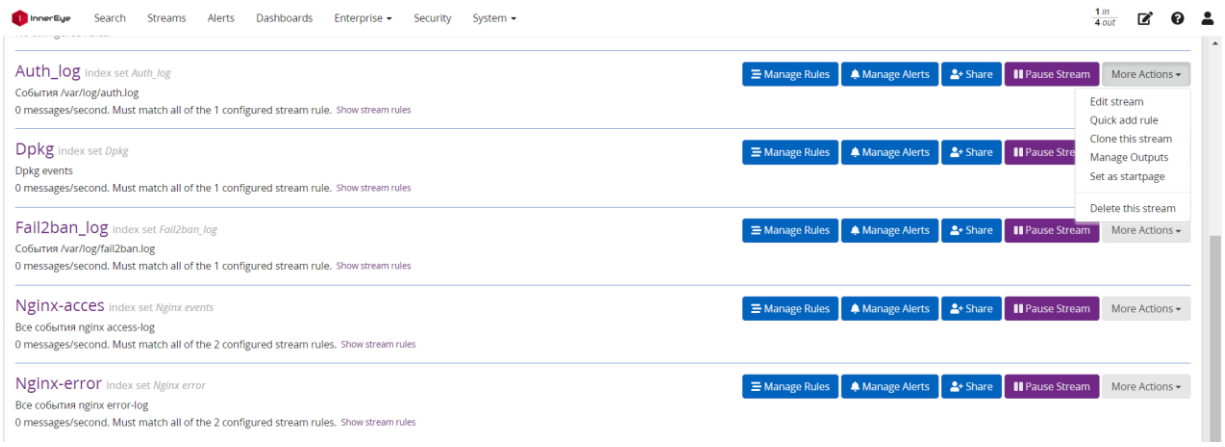


Рис. 4.28. Загальна інформація про потоки та можливі дії з ними

Розподіл подій за потоками складає основу правил.

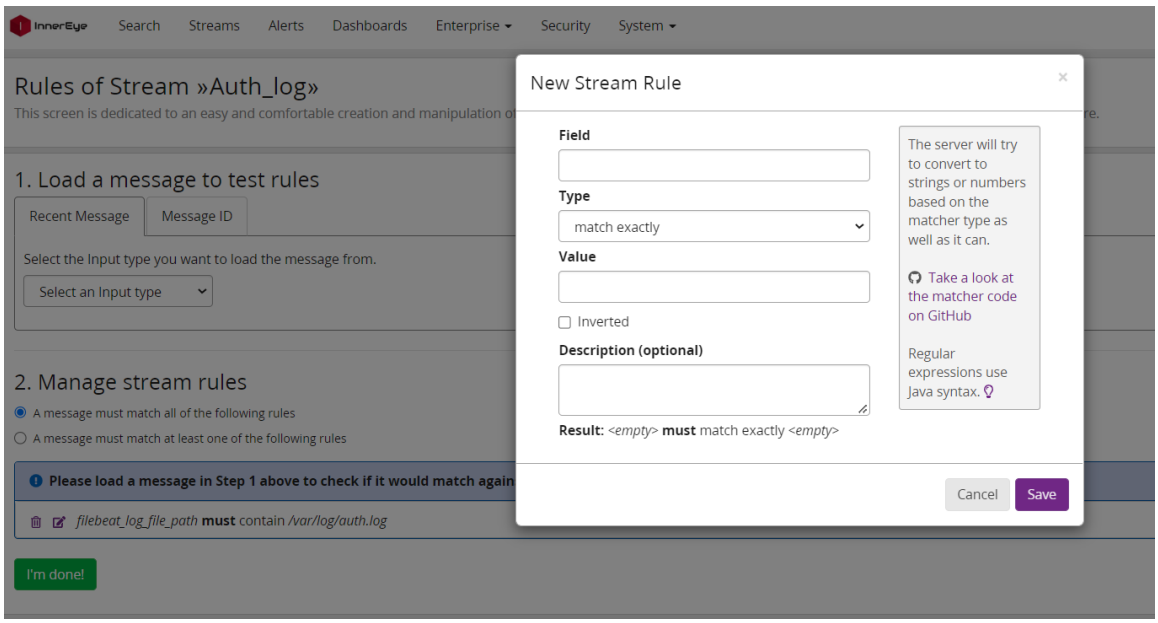


Рис. 4.29. Створення (редагування) правил для розподілу подій на потоки

Використання потоків дозволяє в реальному часі обробляти, сповіщати та пересилати події в інші системи, наприклад, відправляти інформацію про помилки бази даних в іншу систему [18-19].

Обробка подій – конвеєри (pipelines) та правила обробки

Конвеєри – це центральна концепція, що поєднує етапи обробки, що застосовуються до подій. Конвеєри містять правила і можуть бути підключені до одного або кількох потоків, що дозволяють точно контролювати обробку, що застосовується до подій (Рис. 4.30).

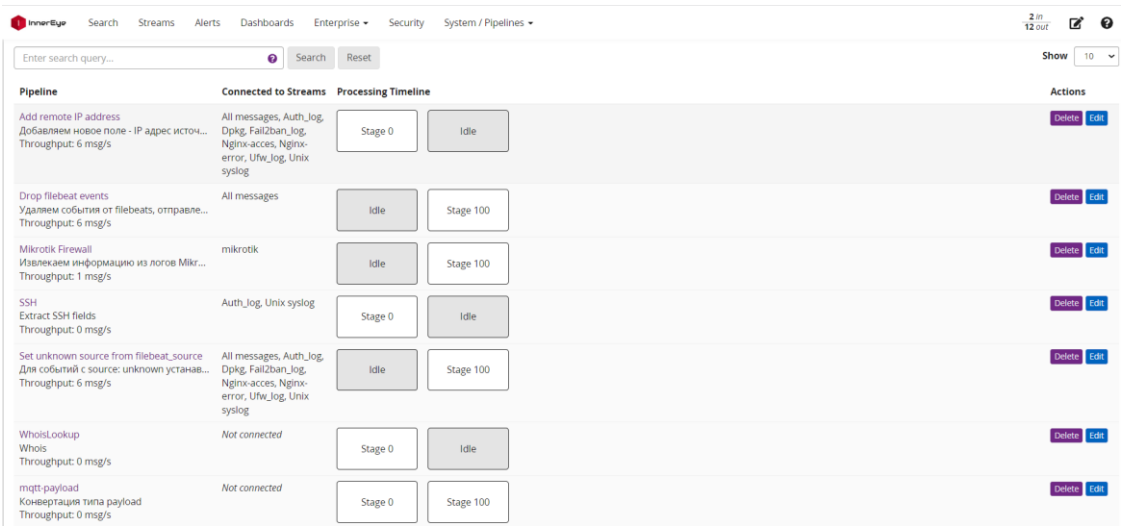


Рис. 4.30. Загальна інформація про конвеєри

Правила обробки – це умови, за якими слідує список дій, і самі по собі вони не мають потоку управління. Тому конвеєри мають ще одне поняття – етапи, які є групами умов та дій, що повинні виконуватися за порядком.

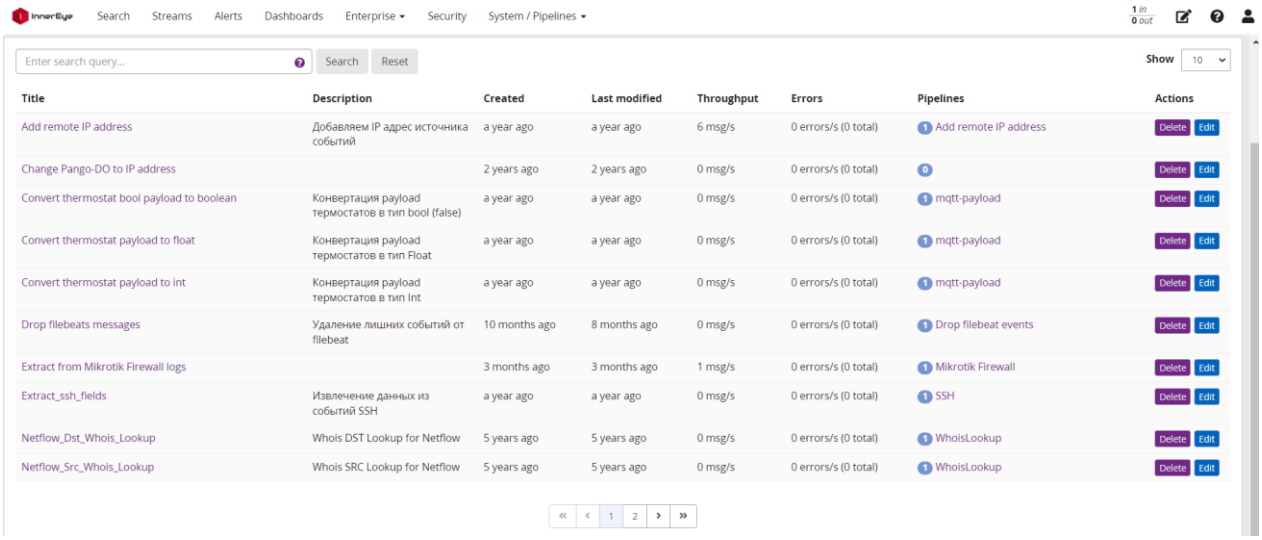


Рис. 4.31. Загальна інформація про правила обробки

Усі етапи з однаковим пріоритетом виконуються одночасно у всіх підключених конвеєрах. Етапи забезпечують необхідний потік управління для прийняття рішення про те, чи слід запускати етапи, що залишилися, в конвеєрі.

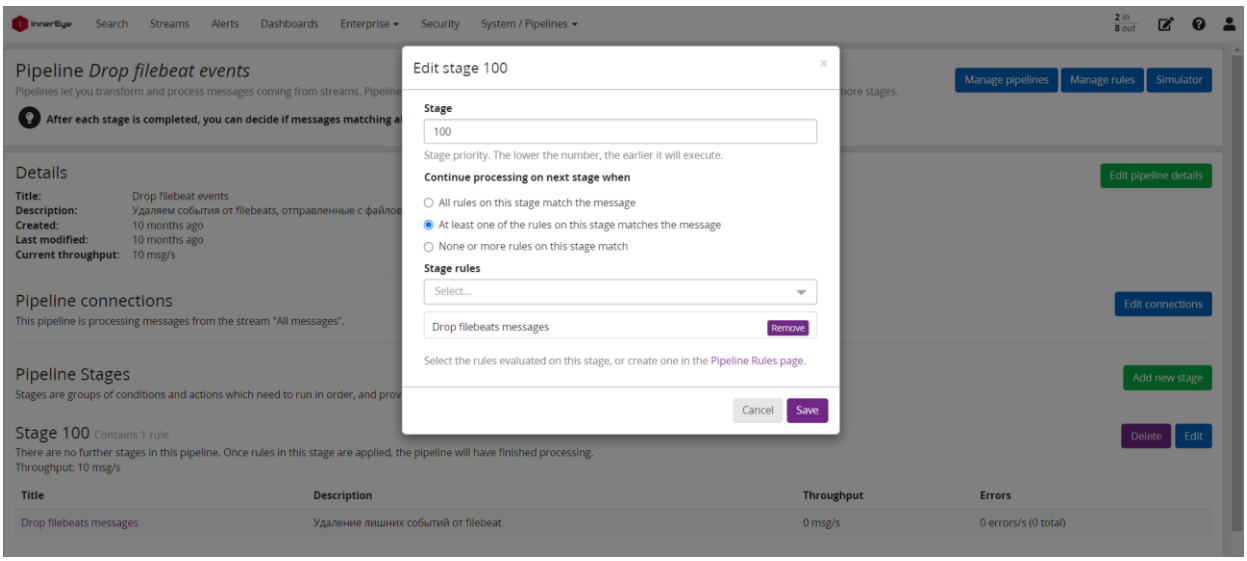


Рис. 4.32. Загальна інформація про етапи обробки та створення-редагування етапу обробки

Етапи виконуються в порядку їхнього пріоритету і не мають інших назв. Пріоритети етапів можуть бути будь-якими цілими числами, позитивними чи негативними. Порядок, що ґрунтується на пріоритеті етапу, дає можливість запускати певні правила до або після інших, які можуть існувати в інших підключених конвеєрах, без зміни цих інших підключених конвеєрів (Рис. 4.33) [5-6].

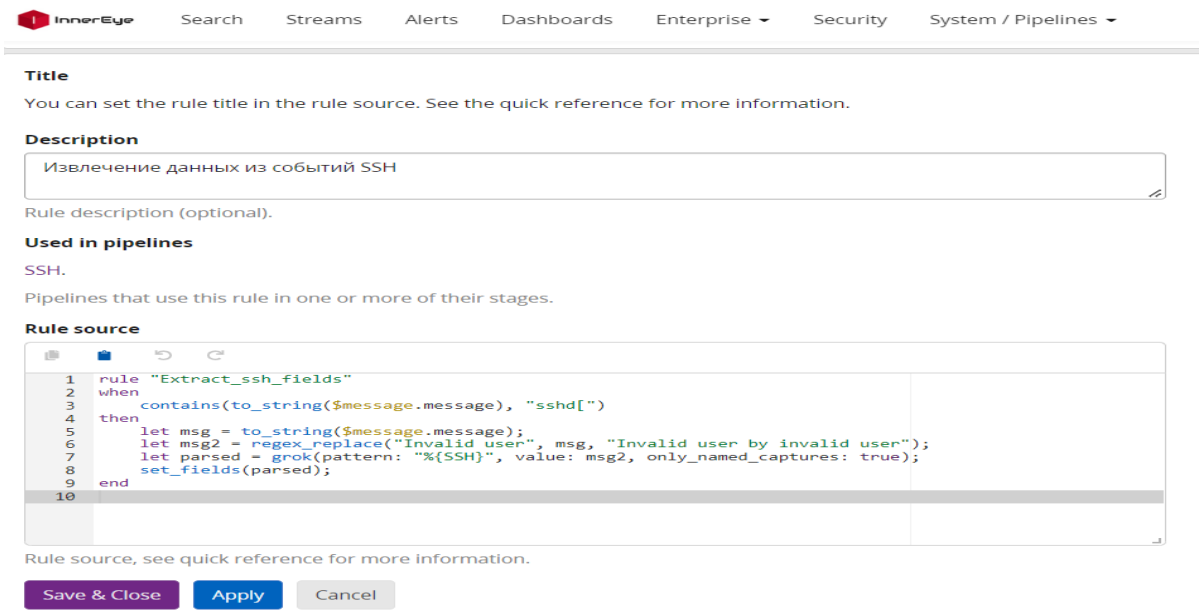


Рис. 4.33. Створення та редагування правил обробки

Правила є основою конвеєрів обробки. Вони містять логіку про те, як змінювати, доповнювати, маршрутизувати та видаляти повідомлення. Обробка повідомлень виконується у функціях. Система корелювання подій та управління IT-інцидентами містить велику кількість вбудованих функцій, що забезпечують перетворення даних, маніпулювання рядками, вилучення даних за допомогою таблиць пошуку, синтаксичний аналіз JSON, тощо. Правила посилаються на імена і тому вони можуть спільно використовуватися багатьма різними конвеєрами. Мета полягає в тому, щоб уможливити створення повторно використовуваних стандартних блоків, спрощуючи обробку даних, характерних для конкретного варіанту використання.

Події та оповіщення

Подія – це умова, яка зіставляє повідомлення, що надходять від джерел (у потоці повідомлень) з періодом часу або агрегацією. Події можна використовувати для угруповання схожих полів, зміни вмісту поля або створення нового вмісту поля для використання з попередженнями та правилами кореляції.

Створення (редагування) події складається з кількох етапів. На першому етапі визначаються загальні властивості події (Рис. 4.34) [5, 18].

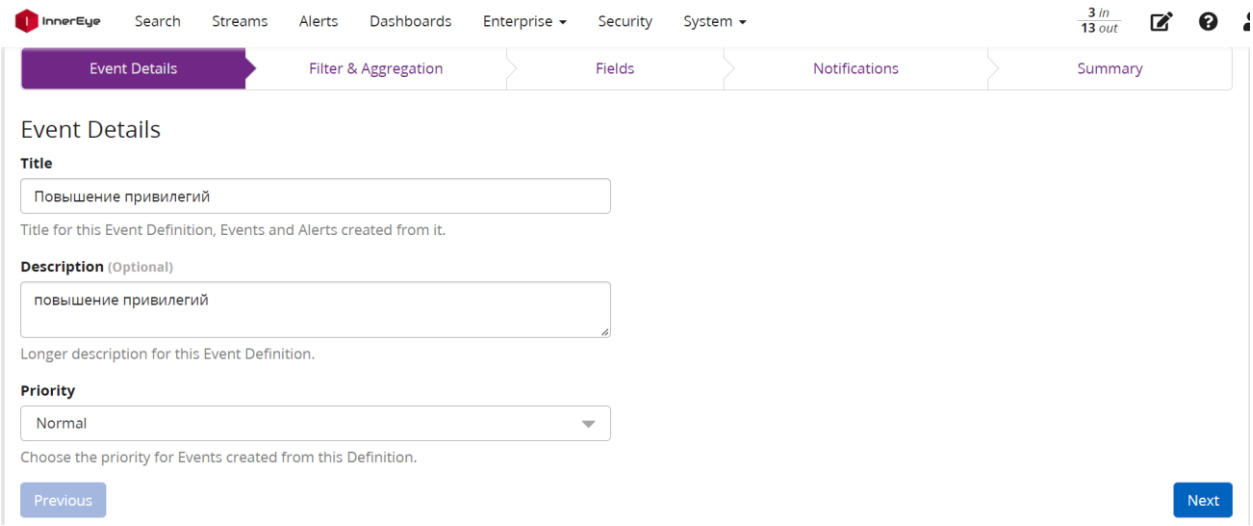


Рис. 4.34. Редагування (створення) події – загальні властивості

На другому етапі можна конкретно описати критерії виявлення події на основі фільтра та агрегації (Рис. 4.35). Фільтр визначається за допомогою пошуку. Для обмеження області пошуку можна вибрати потік, у якому потрібно знайти повідомлення. Можна визначити період часу, протягом якого фільтр шукатиме повідомлення у зворотному напрямку. Пошук буде виконуватись із заданим інтервалом.

На наступному етапі виконується створення полів, що налаштовуються (Рис. 4.36), що дозволяє заповнювати дані з вихідного журналу в індекс подій. Це позбавляє оператора необхідності виконувати наступні пошуки для отримання важливої інформації, і також дають можливість використовувати їх для обмеження обсягу даних, що надсилаються в ціль повідомлень. Подія буде

записана в потік «Всі події» і міститиме поле користувача, а також результат агрегування, що викликало подію.

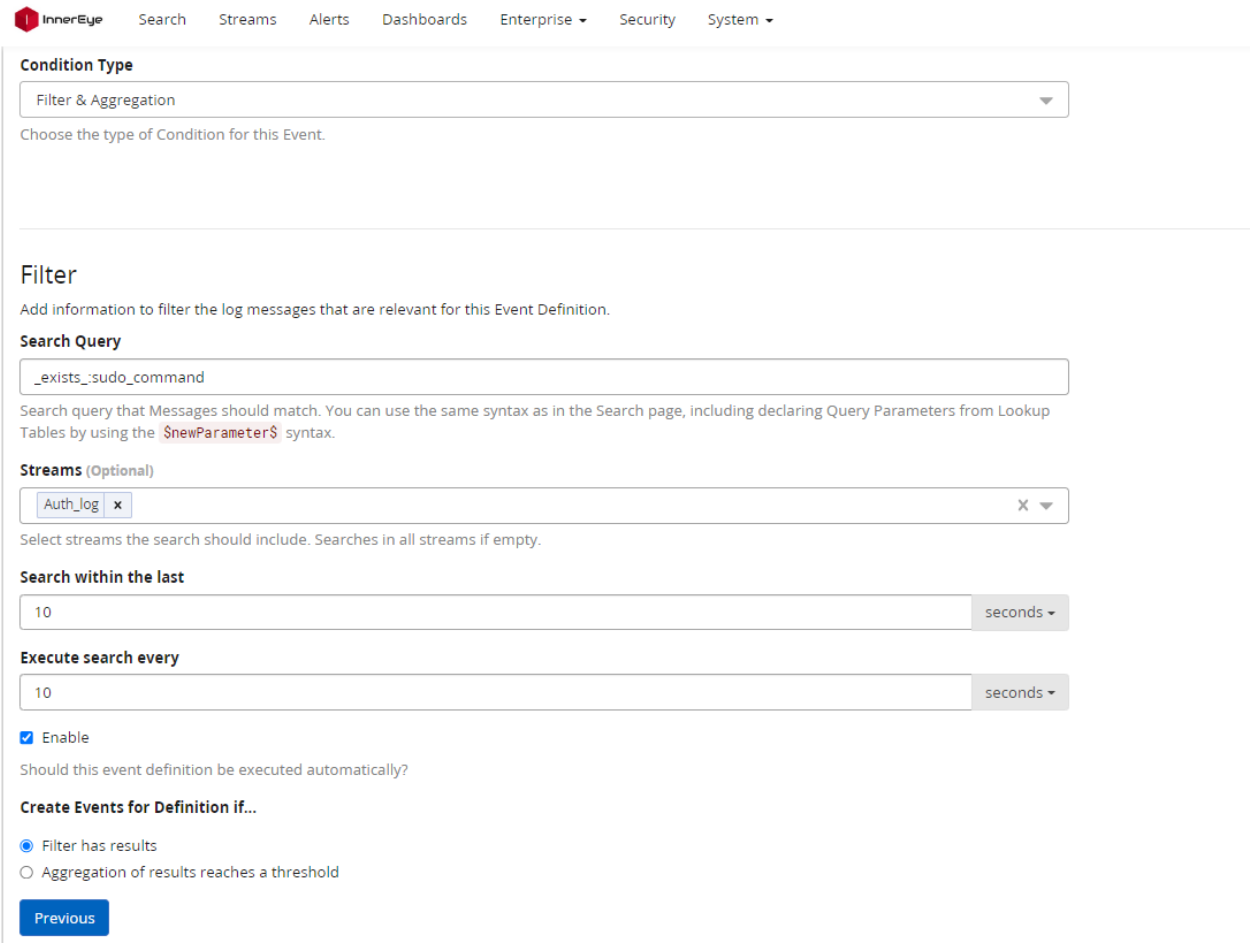


Рис. 4.35. Редагування (створення) події – фільтр

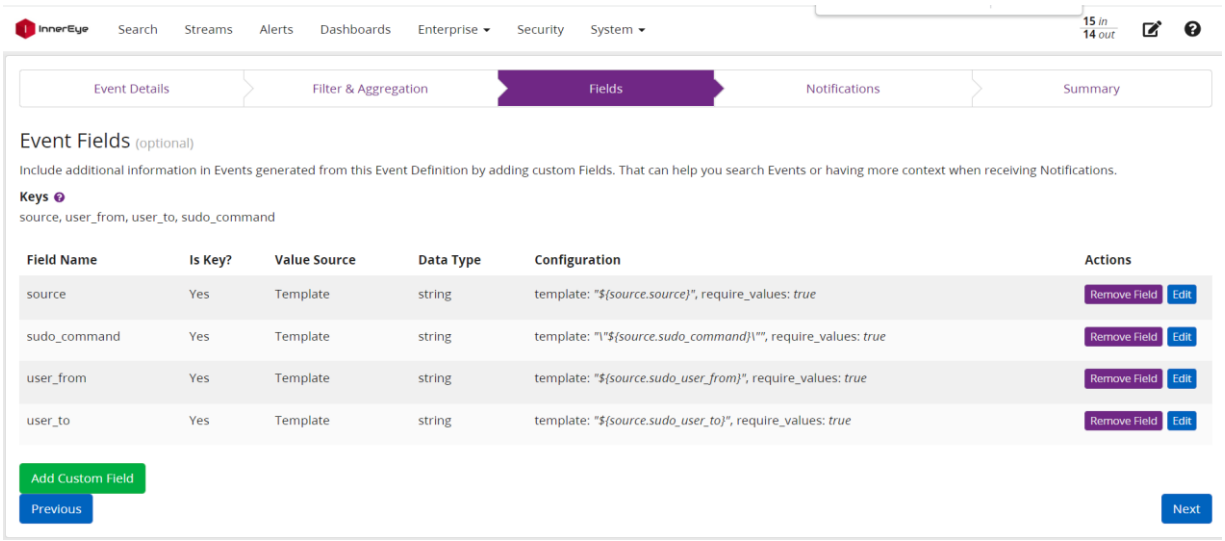


Рис. 4.36. Редагування (створення) події – поля, що налаштовується

На наступному етапі можливе підключення до події сповіщення, при цьому статус події підвищується до оповіщення (Рис. 4.37).

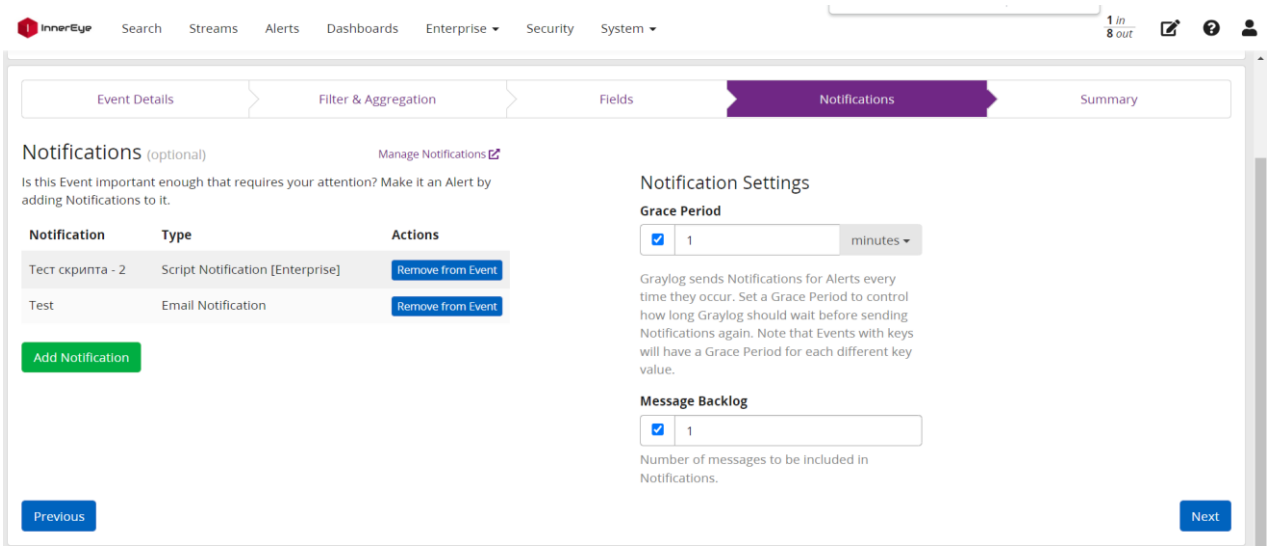


Рис. 4.37. Редагування (створення) події – підключення сповіщень

На останньому етапі здійснюється контроль параметрів та збереження конфігурації події (Рис. 4.38).

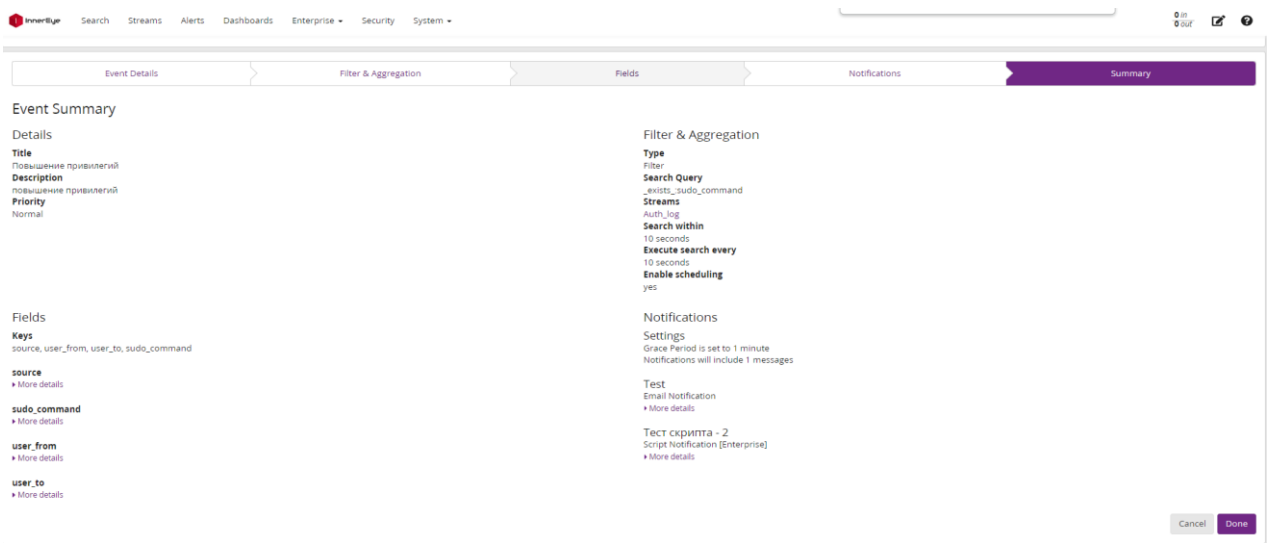


Рис. 4.38. Редагування (створення) події – збереження налаштувань

На сторінці Alerts&Events є повна інформація про всі події та всі оповіщення (Рис. 4.39).

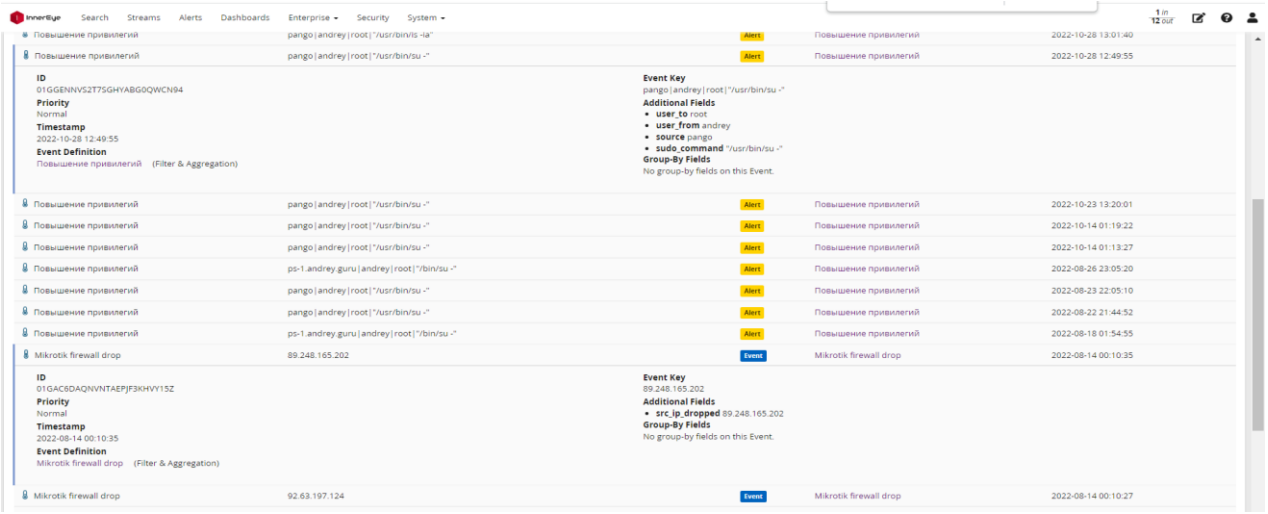


Рис. 4.39. Інформація про події та оповіщення

Інформаційні панелі (dashboards)

Використання інформаційних панелей дозволяє створювати наперед визначені (зумовлені) пошуки за даними. Це дозволяє отримати доступ до важливої інформації в один клік. Інформаційні панелі дозволяють визначити конкретні критерії пошуку, такі як запит або часовий діапазон. Інформаційні панелі (Рис. 4.40) також дозволяють створювати кілька вкладок для різних варіантів використання, відображати результат у повноекранному режимі [5].

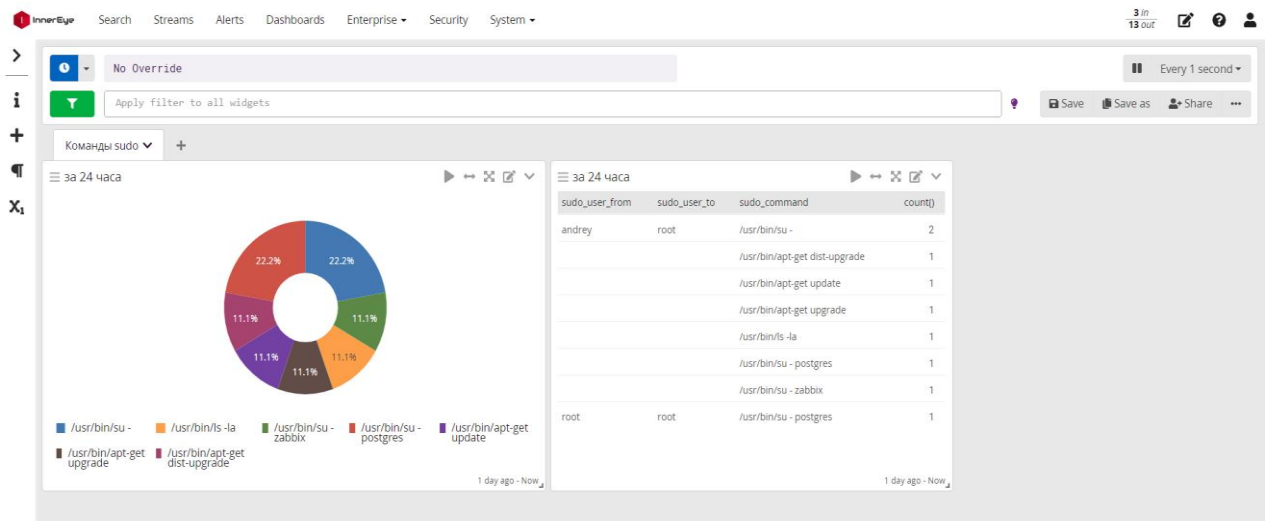


Рис. 4.40. Інформаційна панель – стандартне відображення

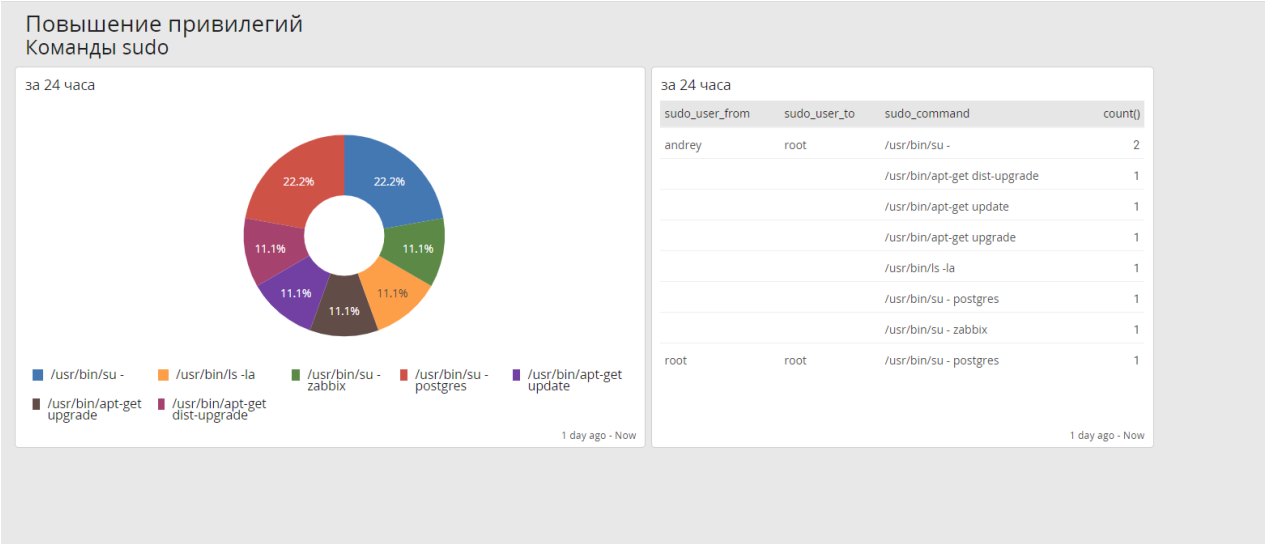


Рис. 4.41. Інформаційна панель – повноекранне відображення

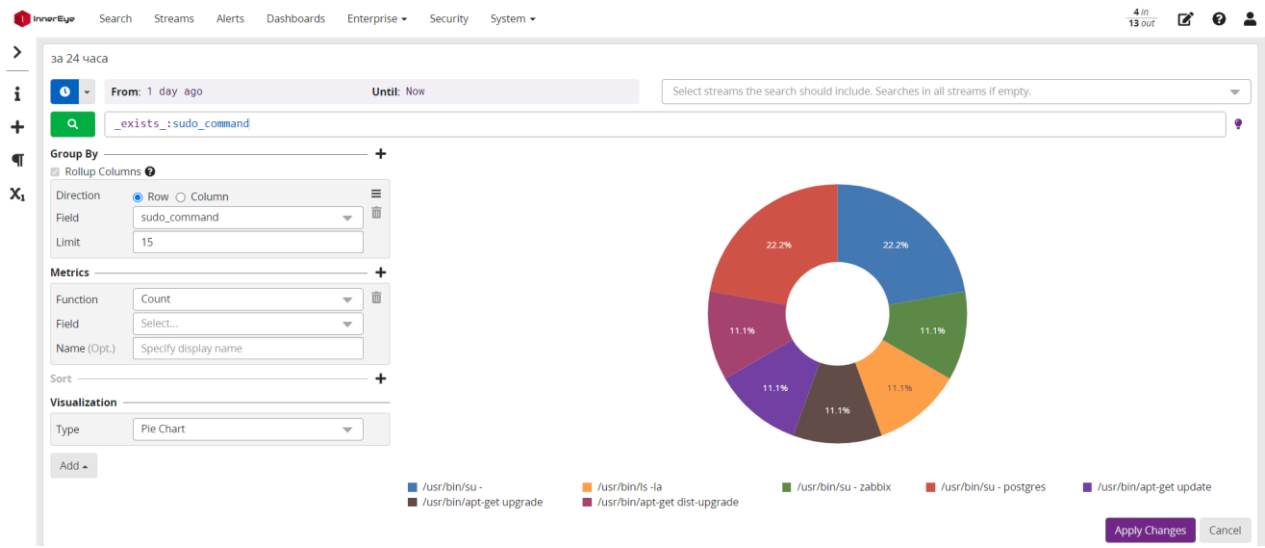


Рис. 4.42. Створення (редагування) елемента інформаційної панелі

Пошук

Сторінка пошуку – це «серце» використання системи корелювання подій та управління IT-інцидентами. На вкладці (Рис. 4.43) можна виконати пошук (запит) та візуалізувати результат за допомогою різних віджетів.

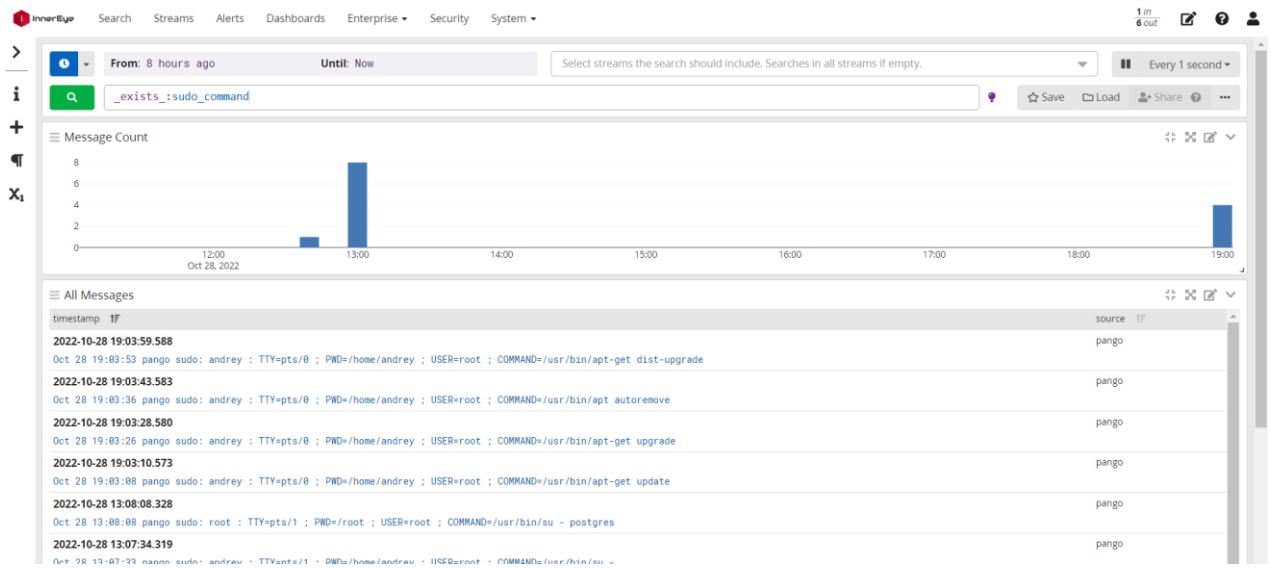


Рис. 4.43. Вікно пошуку – загальний вигляд

Будь-який пошук можна зберегти або експортувати до інформаційної панелі.

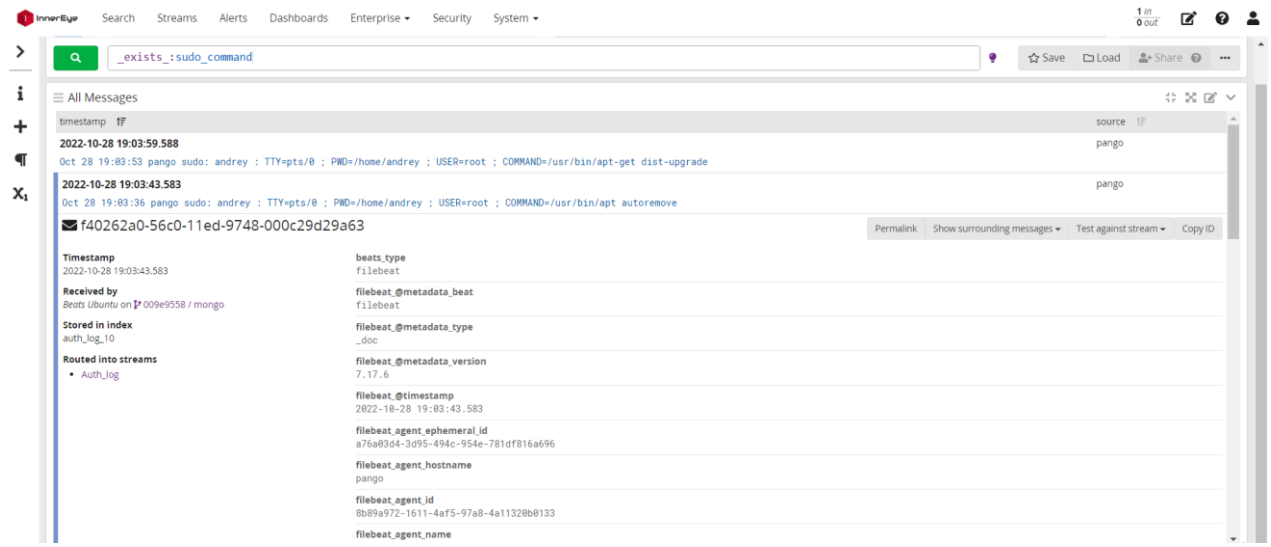


Рис. 4.44. Вікно пошуку – деталі подій

Збережені пошуки дають змогу легко повторно використовувати певні конфігурації пошуку. Інформаційні панелі дозволяють виконувати пошукові запити, специфічні для віджетів, і можуть використовуватись спільно, щоб інші користувачі могли використовувати їх у своїх процесах.

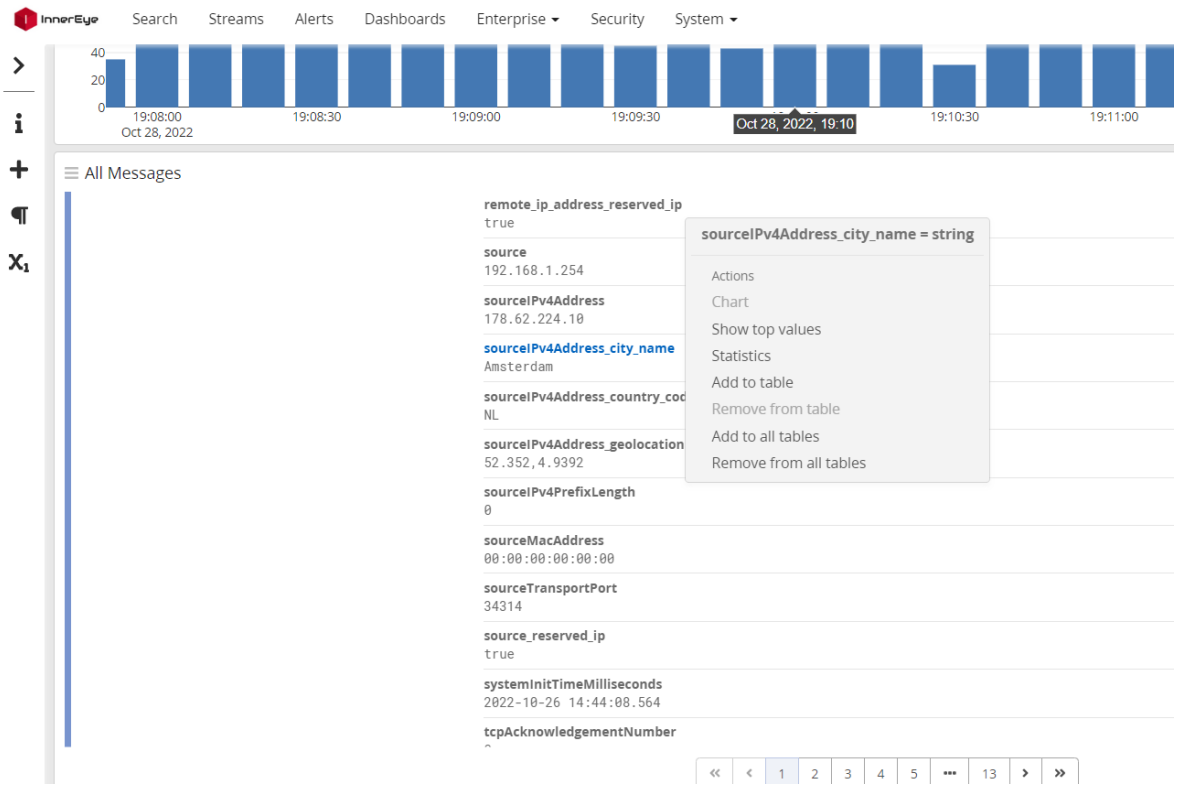


Рис. 4.45. Вікно пошуку – можливі дії

Після завершення роботи з експериментального дослідження системи згідно з розробленою концепцією архітектури проведено навантажувальне випробування макета, яке підтвердило високу ефективність рішень (модулів), що використовуються в розробленій системі корелювання подій та управління IT-інцидентами. Крім цього, за допомогою спеціалізованих засобів проведено перевірку вихідного коду на наявність уразливостей, внаслідок якої критичних уразливостей не було виявлено.

Таким чином, розроблена система дозволяє забезпечувати ефективне корелювання подій та управління інцидентами, які виникають в КІ і мають вплив на КВР. На відміну від існуючих рішень, система враховує взаємозв'язки між подіями, їх часові характеристики та критичність сервісів, що підвищує обґрунтованість прийняття рішень під час реагування на інциденти [5-6].

Для формалізації процесів управління IT-інцидентами на ОКІ в роботі розроблено *інформаційну технологію управління IT-інцидентами* (рис. 4.46), що базується на використанні структурно-аналітичної моделі оброблення даних,

онтологіко-реляційного сховища даних та інтеграційної шини даних. Такий підхід забезпечує інтеграцію різнорідних джерел інформації та узгоджене функціонування компонентів системи.

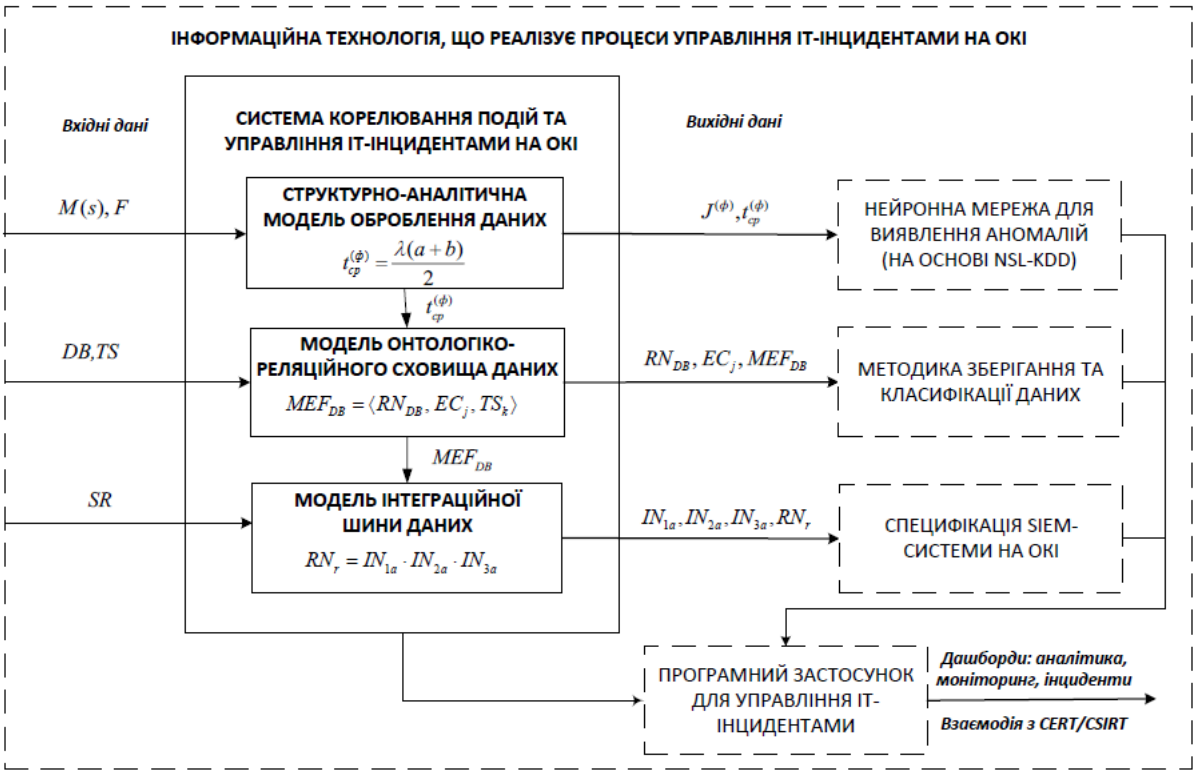


Рис. 4.46. Інформаційна технологія управління ІТ-інцидентами на ОКІ

Вхідними даними інформаційної технології є:

- потоки подій безпеки (журнали, мережеві та системні повідомлення);
- параметри структурно-аналітичної моделі оброблення даних;
- множини баз даних і облікових задач (DB, TS);
- множина сервісів системи (SR), що визначають функціональну структуру ОКІ.

Реалізація зазначених процесів забезпечується:

- нейронною мережею для виявлення аномалій у потоках подій;
- методикою класифікації та зберігання даних;
- специфікацією побудови SIEM-системи;
- спеціалізованим програмним застосунком управління ІТ-інцидентами.

В процесі функціонування системи реалізується послідовність операцій, що включає:

- збирання, індексацію та зберігання подій;
- корелювання подій та їх аналітичне оброблення;
- розрахунок часових характеристик інцидентів;
- визначення рангу критичності сервісів;
- формування керуючих впливів для реагування на інциденти.

Результатом функціонування інформаційної технології є:

- виявлення IT-інцидентів та аномалій;
- визначення пріоритетів їх оброблення;
- формування аналітичних звітів;
- візуалізація результатів моніторингу у вигляді дашбордів;
- забезпечення взаємодії з командами реагування (CERT/CSIRT);
- розроблення рекомендацій щодо реагування та підвищення рівня захищеності ОКІ.

Експериментальне дослідження реалізації зазначеної інформаційної технології в складі системи корелювання подій та управління IT-інцидентами на ОКІ підтвердило її відповідність вимогам міжнародних стандартів і найкращих світових практик створення систем управління IT-інцидентами.

У ході дослідження інформаційної технології забезпечено:

- централізоване управління компонентами системи;
- візуалізацію даних через відповідні інтерфейси;
- підтримку відкритого програмного інтерфейсу (API);
- реалізацію механізмів аутентифікації та авторизації;
- масштабованість та відмовостійкість системи;
- ефективний збір, фільтрацію та оброблення подій;
- управління обліковими записами користувачів.

Розроблена система корелювання подій та управління IT-інцидентами на ОКІ може використовуватись для управління інцидентами, які виникають в КІ і мають вплив на КВР.

4.5. Висновки до четвертого розділу

Таким чином, у цьому розділі на основі запропонованої інформаційної технології та розробленого програмного застосунку проведено експериментальне дослідження і верифіковано отримані у роботі моделі та систему корелювання подій та управління ІТ-інцидентами на ОКІ. Крім того, доведено можливість управління ІТ-інцидентами на ОКІ на основі розроблених та удосконалених моделей і їх синтезу у систему управління ІТ-інцидентами.

4.6. Список літератури до четвертого розділу

1. Gnatyuk S., Satybaldiyeva F., Sydorenko V., Zhyharevych O., Polozhentsev A. «Model of Information Technology for Efficient Data Processing in Cloud-based Malware Detection Systems of Critical Information Infrastructure», CEUR Workshop Proceedings, 2023, vol. 3421, pp. 206-213.

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

3. Melnyk V., Bahnyuk N., Melnyk K., Zhyharevych O., Panasyuk N. «Implementation of the simplified communication mechanism in the cloud of high performance computations». East-European journal of Enterprise Technologies. Kharkiv, 2017. No 2/2/86. P. 24-32.

4. Melnyk, V., Pekh, P., Melnyk, K., Bahnyuk, N., Zhyharevych, O. «Design and implementation of interdomain communication mechanism for high performance data processing». East-European journal of Enterprise Technologies. Kharkiv, 2016. No 1(9). P. 10-15.

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

6. Berdibayev R., Gnatyuk S., Tynymbayev S., Sydorenko V. «Advanced technologies of cyber incident management in critical infrastructure». Kyiv: Pro Format, 2022, 125 p.

7. Vipin K. K. «Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset». International Journal of Soft Computing and Engineering. 2013. Vol. 3. pp. 332-340.

8. Buyya R., Ranjan R., Calheiros R. «Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities». International Conference on High Performance Computing Simulation USA: IEEE, 2009. pp. 1-11.

9. Oksiiuk O., Chaikovska V., Fesenko A. «Security technique for authentication process in the cloud environment», 2019 IEEE International Scientific-Practical Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 379–382. DOI: 10.1109/PICST47496.2019.9061248.

10. Жигаревич О.К., Бердибаєв Р.Ш., Сидоренко В.М., Положенцев А.А., Кримська А.О. «Модель онтоголіко-реляційного сховища даних для функціонування хмарної SIEM-системи», Проблеми інформатизації та управління. 2023. Т. 4. № 76. С. 17-27.

11. Гнатюк С.О., Бердибаєв Р.Ш., Богун А.М., Сидоренко В.М., Положенцев А.А., Жигаревич О.К. «Інтеграційна шина даних для ефективного функціонування системи управління подіями інформаційної безпеки», Проблеми інформатизації та управління. 2023. Т. 3. № 75. С. 29-40.

12. Gnatyuk S., Berdibayev R., Fesenko A., Kyryliuk O., Bessalov A. «Modern SIEM Analysis and Critical Requirements Definition in the Context of Information Warfare». CEUR Workshop Proceedings, 2021, vol. 3188, pp. 149-166.

13. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. «A concept of the architecture and creation for SIEM system in critical infrastructure». Studies in Systems, Decision and Control, Vol. 346, 2021, pp. 221-242.

14. Gnatyuk S., Berdibayev R., Azarov I., Baisholan N., Lozova I. «Modern Types of Databases for SIEM System Development». CEUR Workshop Proceedings, 2021, vol. 3187, pp. 127-138.

15. Gnatyuk, S., Berdibayev, R., Sydorenko, V., Polozhentsev, A., Ryabyy, M. «Enterprise Service Bus Construction in SOA Architecture for SIEM

Implementation in Critical Information Infrastructure» CEUR Workshop Proceedings, 2022, Vol. 3288, Paper 2, P. 11-20.

16. Gnatyuk S., Berdibayev R., Sydorenko V., Berdibayeva G., Yudin O. «Methodological Bases of Critical Information Infrastructure Identification and Security Assessment», Monograph, Kyiv, «Pro Format» Publishing House, 2023, 129 p.

17. Sydorenko, V., Zhyharevych, O., Berdybaev, R., Polozhentsev, A., Fesenko, A. «Ontological-Relational Data Store Model for a Cloud-based SIEM System Development». CEUR Workshop Proceedings. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-2024), February 28, 2024, Kyiv, Ukraine, 2024. Vol. 3654. P. 343-354.

18. Gnatyuk S., Berdibayev R., Aleksander M., Sydorenko V., Zhyharevych O., Polozhentsev A. «Software System for Cybersecurity Events Correlation and Incident Management in Critical Infrastructure». Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies. 2024. Vol. 213. P. 247-269. Springer, Cham.

19. Polozhentsev A., Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O. «Novel Cyber Incident Management System for 5G-based Critical Infrastructures». IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 2023, P. 1037-1041.

ВИСНОВКИ

Результатом виконаної дисертаційної роботи є розв’язання актуальної науково-технічної задачі розроблення системи корелювання подій та управління ІТ-інцидентами на ОКІ.

У процесі виконання дисертаційної роботи отримані такі наукові та практичні результати:

1. Проведено аналіз сучасних підходів до управління ІТ-інцидентами на ОКІ для виявлення їх переваг та недоліків. За результатами проведеного аналізу підходів до виявлення аномалій в хмарному середовищі встановлено, що кожен метод виявлення має свої переваги та працює краще для певних наборів даних, але жоден не є універсальним і не може виявити всі сто відсотків шкідливих програм. Аналіз існуючих типів баз даних та інтеграційних шин даних, показав, що кожні з них мають свої особливості та суттєві відмінності, які визначають їх сферу використання, а також відрізняються функціоналом, додатковими налаштуваннями та вартістю ліцензії. Крім того, систематизовано та представлено детальний аналіз 16 SIEM-систем за 18 запропонованими критеріями. Зокрема відображено їх функціональність, основний принцип роботи, а також проведено порівняльний аналіз їх можливостей та відмінностей, переваг та недоліків використання, та відповідності до міжнародних специфікацій та стандартів. Проведений аналіз дозволив формалізувати завдання дисертаційного дослідження щодо розроблення моделей та системи корелювання подій та управління ІТ-інцидентами на ОКІ.

2. Удосконалено структурно-аналітичну модель оброблення даних, яка завдяки формулюванню команд для передачі керування програмному клієнту ІКТ, додатковій обробці метаданих у хмарній системі та інтелектуалізованому виявленню сигнатур, дозволяє підвищити ефективність виявлення аномалій в хмарних ІКС. Експериментальне дослідженя моделі дозволило оцінити часові характеристики обробки одного елемента

метаданих та розробити керуючі команди. Її відмінною особливістю є врахування необхідності формування команд передачі управління програмному клієнту ІКС, що загалом підвищило точність результатів оцінки часових характеристик до 1,7 разів, і характеристик спотворень (затримок) до 4,5 разів. Крім того, на основі дата сету NSL-KDD навчено нейронну мережу з точки зору виявлення аномалій типу DoS, U2R, R2L та Probe.

3. Розроблено модель онтологіко-реляційного сховища даних, яка за рахунок попередньої індексації та синтезу двох різних баз даних (Elasticsearch та MongoDB) з відповідними характеристиками, дає можливість покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість пошуку та отримання великих обсягів інформації. Крім того, створено методику, яка дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірності та забезпечувати високу швидкість пошуку.

4. Удосконалено модель інтеграційної шини даних, яка за рахунок декомпозиції функціональності сервісів (кожен з яких відповідає за окреме завдання і може працювати ізольовано від інших) та визначення критичності сервісів, дозволяє розподілити навантаження на послуги та гарантує безперервність обміну даними для ефективного функціонування системи корелювання подій та управління ІТ-інцидентами. На основі розробленої моделі, було сформовано відповідну специфікацію реалізації SIEM-систем на ОКІ, у вигляді основних та додаткових вимог.

5. Отримала подальший розвиток система корелювання подій та управління ІТ-інцидентами на ОКІ, яка за рахунок використання розроблених моделей обробки даних, онтологіко-реляційного сховища даних та інтеграційної шини даних, дає змогу формалізувати інформаційну технологію, що реалізує процеси управління ІТ-інцидентами на ОКІ згідно вимог міжнародних стандартів та найкращих світових практик щодо створення систем управління ІТ-інцидентами. Крім того, розроблено

спеціальний програмний застосунок, який можна використовувати для управління IT-інцидентами, які виникають в КІ і мають вплив на КВР.

6. На основі запропонованої інформаційної технології з використанням розробленого спеціалізованого програмного застосунку, проведено експериментальне дослідження і верифіковано отримані у роботі моделі та систему. Результати дисертації впроваджені і використовуються у діяльності ТОВ «AxxonSoft» (акт про впровадження від 11.03.2026), НДЛ протидії кіберзагрозам авіаційної галузі КАІ (акт про впровадження від 12.02.2024), а також у навчальному процесі кафедри комп'ютерних наук та кібербезпеки Волинського національного університету імені Лесі Українки для підвищення ефективності підготовки фахівців з IT (акт про впровадження від 21.12.2023).

Додаток А. Документи, що підтверджують впровадження результатів дисертації

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«АККСОН СОФТ»
Адреса для листування:
04119, Україна, м. Київ, вул. Дегтярівська, буд. 25-А, корп. 1, оф. 414
Код ЄДРПОУ 37818589
IBAN UA053348510000000026008260738,
в АТ «ПУМБ» м. Київ, МФО 334851
https://ua.axxonsoft.com/
тел.+38044 333-71-90, моб. +38 068 333 71 90

АКТ ВПРОВАДЖЕННЯ

результатів дисертаційної роботи **Жигаревич Оксани Костянтинівни** на тему: «Система корелювання подій та управління IT-інцидентами на об'єктах критичної інфраструктури» подану на здобуття наукового ступеня кандидата технічних наук

Результати наукового дослідження впроваджені у діяльність ТОВ «АККСОН СОФТ». У практичній діяльності підприємства використано розроблену систему корелювання подій та управління IT-інцидентами, що базується на застосуванні моделей оброблення даних, онтологіко-реляційного сховища даних та інтеграційної шини даних.

Запропоновані рішення дозволяють формалізувати інформаційну технологію управління IT-інцидентами на об'єктах критичної інфраструктури та забезпечують її реалізацію відповідно до вимог міжнародних стандартів і найкращих світових практик створення систем управління IT-інцидентами.

Крім того, апробовано розроблений спеціалізований програмний застосунок, який може використовуватися для управління IT-інцидентами, що виникають на об'єктах критичної інфраструктури та впливають на критично важливі ресурси.

Директор
ТОВ «АККСОН СОФТ» **Курінний О.В.**

від 12.02.2024р

АКТ

впровадження у науково-дослідну діяльність
результатів дисертаційної роботи Жигаревич Оксани Костянтинівни
«Система корелювання подій та управління IT-інцидентами на об'єктах критичної інфраструктури» на здобуття наукового ступеня кандидата технічних наук

Комісія у складі: голова –провідний науковий співробітник Науково-дослідної лабораторії протидії кіберзагрозам в авіаційній галузі к.т.н., ст. дослідник Охріменко Т.О., старший науковий співробітник, к.т.н., Ковтун М.Г., молодший науковий співробітник Поліщук Ю.Я. склали даний акт про те, що результати дисертаційної роботи Жигаревич Оксани Костянтинівни впровадженні у науково-дослідну діяльність та використовуються в Науково-дослідній лабораторії протидії кіберзагрозам в авіаційній галузі.

№ з/п	Назва роботи, що впроваджується	Форма впровадження	Ефективність від впровадження
	1	2	3
1.	Програмна реалізація структурно-аналітичної моделі оброблення даних.	Матеріали дослідження використовувались під час виконання гранатової програми «Cyber Hygiene E-Learning Course» (grant number G-202112-68299)	Автоматизує процес оброблення даних та дозволяє підвищити ефективність виявлення аномалій в хмарних ІКС. Крім того, на основі дата сету NSL-KDD навчено нейронну мережу з точки зору виявлення аномалій типу DoS, U2R. R2L та Probe;
2.	Програмна реалізація системи корелювання подій та управління IT-інцидентами.	Наукова стаття в рамках НДР	Автоматизує роботу системи корелювання подій та управління IT-інцидентами та дає змогу забезпечити управління IT-інцидентами на ОКІ згідно вимог міжнародних стандартів та найкращих світових практик щодо створення систем управління IT-інцидентами.

Голова комісії,

к.т.н., ст. дослідник
п.н.с. НДЛ протидії кіберзагрозам
в авіаційній галузі — Тетяна ОХРІМЕНКО

Члени комісії:

к.т.н., с.н.с НДЛ протидії кіберзагрозам
в авіаційній галузі — Марія КОВТУН

м.н.с НДЛ протидії кіберзагрозам
в авіаційній галузі — Юлія ПОЛІЩУК

ЗАТВЕРДЖУЮ:
Ректор
Волинського національного
університету імені Лесі Українки
_____ Анатолій ЦЬОСЬ
«21» грудня 2023 р.

АКТ

впровадження у навчальний процес результатів дисертаційної роботи Жигаревич Оксани Костянтинівни «Система корелювання подій та управління ІТ-інцидентами на об'єктах критичної інфраструктури» на здобуття наукового ступеня кандидата технічних наук.

Комісія у складі: голова – завідувач кафедри комп'ютерних наук та кібербезпеки Гришанович Т.О., професор кафедри комп'ютерних наук та кібербезпеки Пастернак Я.М., доцент кафедри комп'ютерних наук та кібербезпеки Булатецька Л.В. склали даний акт про те, що результати дисертаційної роботи Жигаревич Оксани Костянтинівни впровадженні у навчальний процес та використовуються на кафедрі комп'ютерних наук та кібербезпеки у 2022-2023 навчальному році при викладанні дисципліни «Корпоративна безпека».

№ з/п	Назва роботи, що впроваджується	Форма впровадження	Ефективність від впровадження
	1	2	3
1.	Класифікація сучасних підходів до управління ІТ-інцидентами на об'єктах критичної інфраструктури.	Лекція	Ознайомлення студентів з сучасними підходами до управління ІТ-інцидентами на об'єктах критичної інфраструктури.
2.	Розробка моделі онтологіко-реляційного сховища даних для управління ІТ-інцидентами на об'єктах критичної інфраструктури.	Лабораторне заняття	Ознайомлення студентів з методикою зберігання та класифікації даних, яка дозволяє отримувати доступ до зовнішніх сховищ даних, проводити масштабування, агрегацію, аналіз, збір закономірностей та забезпечувати високу швидкість пошуку.

Голова комісії,
завідувач кафедри комп'ютерних
наук та кібербезпеки — Тетяна ГРИШАНОВИЧ

Члени комісії:
професор кафедри комп'ютерних
наук та кібербезпеки — Ярослав ПАСТЕРНАК

доцент кафедри комп'ютерних наук
та кібербезпеки — Леся БУЛАТЕЦЬКА

Додаток Б. Лістинги (код) програмного за стосунку

==========

HORUSEC ENDED THE ANALYSIS WITH STATUS OF "success" AND WITH THE FOLLOWING RESULTS:

==========

Analysis Started at: 2021-11-15 23:47:11

Analysis Finished at: 2021-11-15 23:49:49

==========

Language: Go

Severity: HIGH

Line: 333

Column: 9

SecurityTool: GoSec

Confidence: MEDIUM

File: /home/andrey/github/innereye/cmd/dtester/main.go

Code: 332: rand.Seed(time.Now().UTC().UnixNano())

333: return rand.Intn(m) + min

334: }

Details: Use of weak random number generator (math/rand instead of crypto/rand)

Type: Vulnerability

ReferenceHash: a7ed37d81102f95c6bc882b4de702c07f043b06cece93386947db09b859c0fcb

==========

Language: Go

Severity: HIGH

Line: 135

Column: 74

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dpluger/main.go

Code: TLSClientConfig = &tls.Config{InsecureSkipVerify: true}

136: }

Details: TLS InsecureSkipVerify set true.

Type: Vulnerability

ReferenceHash: e3bdcdcf4f6415f48bc0abcb89a6ae9b417d3353e1c96145308fdc176c0a9238

==========

Language: Go

Severity: MEDIUM

Line: 430

Column: 12

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dtester/main.go

Code: stamp = time.Now().UTC().Format(time.RFC3339)

430: f, err := os.OpenFile(logfile, os.O_APPEND|os.O_

Details: Potential file inclusion via variable

Type: Vulnerability

ReferenceHash: 4e2fa24b6ca21b95c80591445ab4078b18a94b49e860a349ea705541211b70e7

==========

Language: Go

Severity: MEDIUM

Line: 434

Column: 2

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dtester/main.go

Code: 433: }

434: defer f.Close()

435: vJSON, err := json.Marshal(e)

Details: Deferring unsafe method "Close" on type "*os.File"

Type: Vulnerability

ReferenceHash: e583b9c7f430e2bce0154262dd91a3c0e7e3337ae534114780c4cd07ec06b2bf

==========

Language: Go

Severity: MEDIUM

Line: 236

Column: 4

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dsiem/main.go

Code: 235: }

236: defer fo.Close()

237: wrt := bufio.NewWriter(fo)

Details: Deferring unsafe method "Close" on type "*os.File"

Type: Vulnerability

ReferenceHash: 16fc286350cb6136991458d6f990988abf65712125f3b7c0c57b3efcd64b9743

==========

Language: Go

Severity: LOW

Line: 442

Column: 2

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dtester/main.go

Code: 1: }

442: f.SetDeadline(time.Now().Add(60 * time.Second))

443: _, err = f.WriteString(string(vJSO

Details: Errors unhandled.

Type: Vulnerability

ReferenceHash: d6bbb023f825f02be0d579653903622029c5393dfb25190c0a6a1daa5a4866cf

==========

Language: Go

Severity: LOW

Line: 238

Column: 4

SecurityTool: GoSec

Confidence: HIGH

File: /home/andrey/github/innereye/cmd/dsiem/main.go

Code: wrt := bufio.NewWriter(fo)

238: trace.Start(wrt)

239: t := time.NewTimer(10 * time.Second)

Details: Errors unhandled.

Type: Vulnerability

ReferenceHash: 1472fe1fc4cbe59da50ef75673d4af577d546ac120a15294643e76b68eb682f9

==========

In this analysis, a total of 7 possible vulnerabilities were found and we classified them into:

Total of Vulnerability HIGH is: 2

Total of Vulnerability MEDIUM is: 3

Total of Vulnerability LOW is: 2

==========

HORUSEC ENDED THE ANALYSIS WITH STATUS OF "success" AND WITH THE FOLLOWING RESULTS:

==========

Analysis StartedAt: 2021-11-15 23:54:04

Analysis FinishedAt: 2021-11-15 23:56:45

==========

Language: JavaScript

Severity: CRITICAL

Line: 12797

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: xmlhttprequest-ssl

Details: The xmlhttprequest-ssl package before 1.6.1 for Node.js disables SSL certificate validation by default, because rejectUnauthorized (when the property exists but is undefined) is considered to be false within the https.request function of Node.js. In other words, no certificate is ever rejected.

Type: Vulnerability

ReferenceHash: 015157805a0003b1d55cf6fe0e1b16a2b39bef178e366f53b4fc1b81ebfd3520

==========

Language: Generic

Severity: CRITICAL

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: serialize-javascript:2.1.2

Details: The application deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

Type: Vulnerability

ReferenceHash: 3e6854a9649f9a6fde9cdb8e33e7f2409498a88ee92a9340d9baf5c4c21e8780

==========

Language: JavaScript

Severity: HIGH

Line: 2078

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ansi-html

Details: This affects all versions of package ansi-html. If an attacker provides a malicious string, it will get stuck processing the input for an extremely long time.

Type: Vulnerability

ReferenceHash: 047cf56f31da1135b9fb27d20714fe8f06de8e5b2c6f3ac4bef81d6ae2a6e0c9

==========

Language: JavaScript

Severity: HIGH

Line: 8356

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: node-forge

Details: The package node-forge before 0.10.0 is vulnerable to Prototype Pollution via the util.setPath function. Note: Version 0.10.0 is a breaking change removing the vulnerable functions.

Type: Vulnerability

ReferenceHash: f2366c8b515aa5fe3743d2e4614333301dd379c875138cdad0a9c696b7940e2a

==========

Language: JavaScript

Severity: HIGH

Line: 5344

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ini

Details: ### Overview

The `ini` npm package before version 1.3.6 has a Prototype Pollution vulnerability.

If an attacker submits a malicious INI file to an application that parses it with `ini.parse`, they will pollute the prototype on the application. This can be exploited further depending on the context.

Patches

This has been patched in 1.3.6

Steps to reproduce

payload.ini
```
[__proto__]
polluted = "polluted"
```
poc.js:
```
var fs = require('fs')
var ini = require('ini')

var parsed = ini.parse(fs.readFileSync('./payload.ini', 'utf-8'))
console.log(parsed)
console.log(parsed.__proto__)
console.log(polluted)
```
```
> node poc.js
{}
{ polluted: 'polluted' }
{ polluted: 'polluted' }
polluted
```
Type: Vulnerability

ReferenceHash: 71f335f10f05195a151febdad59a8e1e96d0932238edc1088bf535eae8f64e40

==========

Language: JavaScript

Severity: HIGH

Line: 138

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: serialize-javascript

Details: serialize-javascript prior to 3.1.0 allows remote attackers to inject arbitrary code via the function "deleteFunctions" within "index.js".

An object such as `{"foo": /1"/, "bar": "a\"@__R-<UID>-0__@"}` was serialized as `{"foo": /1"/, "bar": "a\/1"/}`, which allows an attacker to escape the `bar` key. This requires the attacker to control the values of both `foo` and `bar` and guess the value of `<UID>`. The UID has a keyspace of approximately 4 billion making it a realistic network attack.

Type: Vulnerability

ReferenceHash: 4e868ce9207b8a8dfb552ad2d03c2d4c98e0a0c79513fac7f6c3c5240049aa4f

==========

Language: JavaScript

Severity: HIGH

Line: 5651

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

node-tar aims to guarantee that any file whose location would be modified by a symbolic link is not extracted. This is, in part, achieved by ensuring that extracted directories are not symlinks. Additionally, in order to prevent unnecessary stat calls to determine whether a given path is a directory, paths are cached when directories are created.

This logic was insufficient when extracting tar files that contained two directories and a symlink with names containing unicode values that normalized to the same value. Additionally, on Windows systems, long path portions would resolve to the same file system entities as their 8.3 "short path" counterparts. A specially crafted tar archive could thus include directories with two forms of the path that resolve to the same file system entity, followed by a symbolic link with a name in the first form, lastly followed by a file using the second form. It led to bypassing node-tar symlink checks on directories, essentially allowing an untrusted tar file to symlink into an arbitrary location and subsequently extracting arbitrary files into that location, thus allowing arbitrary file creation and overwrite.

The v3 branch of `node-tar` has been deprecated and did not receive patches for these issues. If you are still using a v3 release we recommend you update to a more recent version of `node-tar`. If this is not possible, a workaround is available below.

Patches

6.1.9 || 5.0.10 || 4.4.18

Workarounds

Users may work around this vulnerability without upgrading by creating a custom filter method which prevents the extraction of symbolic links.

```js
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  filter: (file, entry) => {
    if (entry.type === 'SymbolicLink') {
      return false
```

```
    } else {
      return true
    }
  }
})
```

Users are encouraged to upgrade to the latest patched versions, rather than attempt to sanitize tar input themselves.

Fix

The problem is addressed in the following ways, when comparing paths in the directory cache and path reservation systems:

1. The `String.normalize('NFKD')` method is used to first normalize all unicode to its maximally compatible and multi-code-point form.

2. All slashes are normalized to `/` on Windows systems (on posix systems, `\` is a valid filename character, and thus left intact).

3. When a symbolic link is encountered on Windows systems, the entire directory cache is cleared. Collisions related to use of 8.3 short names to replace directories with other (non-symlink) types of entries may make archives fail to extract properly, but will not result in arbitrary file writes.

Type: Vulnerability

ReferenceHash: 8af42859b6ce157ed4d33d9b2376a47b661176d65f79fbfbf6f1f73e23ac1b09

==========

Language: JavaScript

Severity: HIGH

Line: 7584

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: kind-of

Details: Versions of `kind-of` 6.x prior to 6.0.3 are vulnerable to a Validation Bypass. A maliciously crafted object can alter the result of the type check, allowing attackers to bypass the type checking validation.

Recommendation

Upgrade to versions 6.0.3 or later.

Type: Vulnerability

ReferenceHash: a290e7f1a267468b6551be68e850c538f4a462eae52cdf959777b5751f99d291

==========

Language: JavaScript

Severity: HIGH

Line: 5651

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

node-tar aims to guarantee that any file whose location would be outside of the extraction target directory is not extracted. This is, in part, accomplished by sanitizing absolute paths of entries within the archive, skipping archive entries that contain `..` path portions, and resolving the sanitized paths against the extraction target directory.

This logic was insufficient on Windows systems when extracting tar files that contained a path that was not an absolute path, but specified a drive letter different from the extraction target, such as `C:some\path`. If the drive letter does not match the extraction target, for example `D:\extraction\dir`, then the result of `path.resolve(extractionDirectory, entryPath)` would resolve against the current working directory on the `C:` drive, rather than the extraction target directory.

Additionally, a `..` portion of the path could occur immediately after the drive letter, such as `C:../foo`, and was not properly sanitized by the logic that checked for `..` within the normalized and split portions of the path.

This only affects users of `node-tar` on Windows systems.

Patches

4.4.18 || 5.0.10 || 6.1.9

Workarounds

There is no reasonable way to work around this issue without performing the same path normalization procedures that node-tar now does.

Users are encouraged to upgrade to the latest patched versions of node-tar, rather than attempt to sanitize paths themselves.

Fix

The fixed versions strip path roots from all paths prior to being resolved against the extraction target folder, even if such paths are not "absolute".

Additionally, a path starting with a drive letter and then two dots, like `c:../`, would bypass the check for `..` path portions. This is checked properly in the patched versions.

Finally, a defense in depth check is added, such that if the `entry.absolute` is outside of the extraction taret, and we are not in preservePaths:true mode, a warning is raised on that entry, and

it is skipped. Currently, it is believed that this check is redundant, but it did catch some oversights in development.

Type: Vulnerability

ReferenceHash: 24e71ccde94866c63a1fae7253a5df22037f23670610e213d4ab5c81a2830323

==========

Language: JavaScript

Severity: HIGH

Line: 8387

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

`node-tar` aims to prevent extraction of absolute file paths by turning absolute paths into relative paths when the `preservePaths` flag is not set to `true`. This is achieved by stripping the absolute path root from any absolute file paths contained in a tar file. For example `/home/user/.bashrc` would turn into `home/user/.bashrc`.

This logic was insufficient when file paths contained repeated path roots such as `////home/user/.bashrc`. `node-tar` would only strip a single path root from such paths. When given an absolute file path with repeating path roots, the resulting path (e.g. `///home/user/.bashrc`) would still resolve to an absolute path, thus allowing arbitrary file creation and overwrite.

Patches

3.2.2 || 4.4.14 || 5.0.6 || 6.1.1

NOTE: an adjacent issue [CVE-2021-32803](https://github.com/npm/node-tar/security/advisories/GHSA-r628-mhmh-qjhw) affects this release level. Please ensure you update to the latest patch levels that address CVE-2021-32803 as well if this adjacent issue affects your `node-tar` use case.

Workarounds

Users may work around this vulnerability without upgrading by creating a custom `onentry` method which sanitizes the `entry.path` or a `filter` method which removes entries with absolute paths.

```js
const path = require('path')
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  // either add this function...
  onentry: (entry) => {
    if (path.isAbsolute(entry.path)) {
      entry.path = sanitizeAbsolutePathSomehow(entry.path)
      entry.absolute = path.resolve(entry.path)
    }
  },
  // or this one
  filter: (file, entry) => {
    if (path.isAbsolute(entry.path)) {
      return false
    } else {
      return true
    }
  }
})
```

```

Users are encouraged to upgrade to the latest patch versions, rather than attempt to sanitize tar input themselves.

Type: Vulnerability

ReferenceHash: 678695bbc550bd435a98b368adb36e7e67aaced59475a6bc4032df5e639e4644

==========

Language: JavaScript

Severity: HIGH

Line: 5651

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

`node-tar` aims to guarantee that any file whose location would be modified by a symbolic link is not extracted. This is, in part, achieved by ensuring that extracted directories are not symlinks. Additionally, in order to prevent unnecessary `stat` calls to determine whether a given path is a directory, paths are cached when directories are created.

This logic was insufficient when extracting tar files that contained both a directory and a symlink with the same name as the directory. This order of operations resulted in the directory being created and added to the `node-tar` directory cache. When a directory is present in the directory cache, subsequent calls to mkdir for that directory are skipped. However, this is also where `node-tar` checks for symlinks occur.

By first creating a directory, and then replacing that directory with a symlink, it was thus possible to bypass `node-tar` symlink checks on directories, essentially allowing an untrusted tar file to symlink into an arbitrary location and subsequently extracting arbitrary files into that location, thus allowing arbitrary file creation and overwrite.
```

This issue was addressed in releases 3.2.3, 4.4.15, 5.0.7 and 6.1.2.

Patches

3.2.3 || 4.4.15 || 5.0.7 || 6.1.2

Workarounds

Users may work around this vulnerability without upgrading by creating a custom `filter` method which prevents the extraction of symbolic links.

```js
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  filter: (file, entry) => {
    if (entry.type === 'SymbolicLink') {
      return false
    } else {
      return true
    }
  }
})
```

Users are encouraged to upgrade to the latest patch versions, rather than attempt to sanitize tar input themselves.

Type: Vulnerability

ReferenceHash: 1d485e7f7bc55152a77997f1eb401494b3a46193eb169c1a3697fbb6b3f982f9

==========

Language: JavaScript

Severity: HIGH

Line: 11803

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tree-kill

Details: Versions of `tree-kill` prior to 1.2.2 are vulnerable to Command Injection. The package fails to sanitize values passed to the `kill` function. If this value is user-controlled it may allow attackers to run arbitrary commands in the server. The issue only affects Windows systems.

Recommendation

Upgrade to version 1.2.2 or later.

Type: Vulnerability

ReferenceHash: e4698cac3d53611277a0c96a2a2aaf29c403d6714de86ff7c622c522e244a621

==========

Language: JavaScript

Severity: HIGH

Line: 5837

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: glob-parent

Details: This affects the package glob-parent before 5.1.2. The enclosure regex used to check for strings ending in enclosure containing path separator.

Type: Vulnerability

ReferenceHash: 6bddc6600f6bc11da216916f63286948b8a314fc535e5447e7ebf35c11cdd7e6

==========

Language: JavaScript

Severity: HIGH

Line: 8387

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

`node-tar` aims to guarantee that any file whose location would be modified by a symbolic link is not extracted. This is, in part, achieved by ensuring that extracted directories are not symlinks. Additionally, in order to prevent unnecessary `stat` calls to determine whether a given path is a directory, paths are cached when directories are created.

This logic was insufficient when extracting tar files that contained both a directory and a symlink with the same name as the directory. This order of operations resulted in the directory being created and added to the `node-tar` directory cache. When a directory is present in the directory cache, subsequent calls to mkdir for that directory are skipped. However, this is also where `node-tar` checks for symlinks occur.

By first creating a directory, and then replacing that directory with a symlink, it was thus possible to bypass `node-tar` symlink checks on directories, essentially allowing an untrusted tar file to symlink into an arbitrary location and subsequently extracting arbitrary files into that location, thus allowing arbitrary file creation and overwrite.

This issue was addressed in releases 3.2.3, 4.4.15, 5.0.7 and 6.1.2.

Patches

3.2.3 || 4.4.15 || 5.0.7 || 6.1.2

Workarounds

Users may work around this vulnerability without upgrading by creating a custom `filter` method which prevents the extraction of symbolic links.

```js
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  filter: (file, entry) => {
    if (entry.type === 'SymbolicLink') {
      return false
    } else {
      return true
    }
  }
})
```

Users are encouraged to upgrade to the latest patch versions, rather than attempt to sanitize tar input themselves.

Type: Vulnerability

ReferenceHash: 8f02fa7954f63fb32f394e90a3c8bda5e8e1b05c15cb2f03e428cd0a7a4acc17

==========

Language: JavaScript

Severity: HIGH

Line: 11809

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: trim-newlines

Details: The trim-newlines package before 3.0.1 and 4.x before 4.0.1 for Node.js has an issue related to regular expression denial-of-service (ReDoS) for the .end() method.

Type: Vulnerability

ReferenceHash: 2bea6339876b8041f16e2b6d85660731dae19e73ae85fd25cb734f8e648ebbf1

==========

Language: JavaScript

Severity: HIGH

Line: 12797

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: xmlhttprequest-ssl

Details: This affects the package xmlhttprequest before 1.7.0; all versions of package xmlhttprequest-ssl. Provided requests are sent synchronously (async=False on xhr.open), malicious user input flowing into xhr.send could result in arbitrary code being injected and run.

Type: Vulnerability

ReferenceHash: 123c2cda88242d59644ba391d3535d1e9c62858461946810de083471a2011a48

==========

Language: JavaScript

Severity: HIGH

Line: 5651

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

`node-tar` aims to guarantee that any file whose location would be modified by a symbolic link is not extracted. This is, in part, achieved by ensuring that extracted directories are not symlinks. Additionally, in order to prevent unnecessary stat calls to determine whether a given path is a directory, paths are cached when directories are created.

This logic was insufficient when extracting tar files that contained both a directory and a symlink with the same name as the directory, where the symlink and directory names in the archive entry used backslashes as a path separator on posix systems. The cache checking logic used both `\` and `/` characters as path separators, however `\` is a valid filename character on posix systems.

By first creating a directory, and then replacing that directory with a symlink, it was thus possible to bypass node-tar symlink checks on directories, essentially allowing an untrusted tar file to symlink into an arbitrary location and subsequently extracting arbitrary files into that location, thus allowing arbitrary file creation and overwrite.

Additionally, a similar confusion could arise on case-insensitive filesystems. If a tar archive contained a directory at `FOO`, followed by a symbolic link named `foo`, then on case-insensitive file systems, the creation of the symbolic link would remove the directory from the filesystem, but _not_ from the internal directory cache, as it would not be treated as a cache hit. A subsequent file entry within the `FOO` directory would then be placed in the target of the symbolic link, thinking that the directory had already been created.

These issues were addressed in releases 4.4.16, 5.0.8 and 6.1.7.

The v3 branch of `node-tar` has been deprecated and did not receive patches for these issues. If you are still using a v3 release we recommend you update to a more recent version of `node-tar`. If this is not possible, a workaround is available below.

Patches

4.4.16 || 5.0.8 || 6.1.7

Workarounds

Users may work around this vulnerability without upgrading by creating a custom filter method which prevents the extraction of symbolic links.

```js
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  filter: (file, entry) => {
    if (entry.type === 'SymbolicLink') {
      return false
    } else {
      return true
    }
  }
})
```

Users are encouraged to upgrade to the latest patched versions, rather than attempt to sanitize tar input themselves.

Fix

The problem is addressed in the following ways:

1. All paths are normalized to use `/` as a path separator, replacing `\` with `/` on Windows systems, and leaving `\` intact in the path on posix systems. This is performed in depth, at every level of the program where paths are consumed.

2. Directory cache pruning is performed case-insensitively. This _may_ result in undue cache misses on case-sensitive file systems, but the performance impact is negligible.

Caveat

Note that this means that the `entry` objects exposed in various parts of tar's API will now always use `/` as a path separator, even on Windows systems. This is not expected to cause problems, as `/` is a valid path separator on Windows systems, but _may_ result in issues if `entry.path` is compared against a path string coming from some other API such as `fs.realpath()` or `path.resolve()`.

Users are encouraged to always normalize paths using a well-tested method such as `path.resolve()` before comparing paths to one another.

Type: Vulnerability

ReferenceHash: 4486b5678dc03f7c64d2e65a426be3a4546c1b02a395751fcd0c9ae7ba821009

==========

Language: JavaScript

Severity: HIGH

Line: 5651

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar

Details: ### Impact

Arbitrary File Creation, Arbitrary File Overwrite, Arbitrary Code Execution

`node-tar` aims to prevent extraction of absolute file paths by turning absolute paths into relative paths when the `preservePaths` flag is not set to `true`. This is achieved by stripping the absolute path root from any absolute file paths contained in a tar file. For example `/home/user/.bashrc` would turn into `home/user/.bashrc`.

This logic was insufficient when file paths contained repeated path roots such as `////home/user/.bashrc`. `node-tar` would only strip a single path root from such paths. When given an absolute file path with repeating path roots, the resulting path (e.g. `///home/user/.bashrc`) would still resolve to an absolute path, thus allowing arbitrary file creation and overwrite.

Patches

3.2.2 || 4.4.14 || 5.0.6 || 6.1.1

NOTE: an adjacent issue [CVE-2021-32803](https://github.com/npm/node-tar/security/advisories/GHSA-r628-mhmh-qjhw) affects this release level. Please ensure you update to the latest patch levels that address CVE-2021-32803 as well if this adjacent issue affects your `node-tar` use case.

Workarounds

Users may work around this vulnerability without upgrading by creating a custom `onentry` method which sanitizes the `entry.path` or a `filter` method which removes entries with absolute paths.

```js
const path = require('path')
const tar = require('tar')
tar.x({
  file: 'archive.tgz',
  // either add this function...
  onentry: (entry) => {
    if (path.isAbsolute(entry.path)) {
      entry.path = sanitizeAbsolutePathSomehow(entry.path)
      entry.absolute = path.resolve(entry.path)
    }
  },
  // or this one
```

```
  filter: (file, entry) => {
    if (path.isAbsolute(entry.path)) {
      return false
    } else {
      return true
    }
  }
})
```

Users are encouraged to upgrade to the latest patch versions, rather than attempt to sanitize tar input themselves.

Type: Vulnerability

ReferenceHash: 2251aa051751ecc0f36f33cc7bba3803592d818db411a3ed26c272011cef170a

==========

Language: JavaScript

Severity: HIGH

Line: 10908

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: socket.io-parser

Details: The `socket.io-parser` npm package before versions 3.3.2 and 3.4.1 allows attackers to cause a denial of service (memory consumption) via a large packet because a concatenation approach is used.

Type: Vulnerability

ReferenceHash: 7eddeab74aa3f7e208cae2f08c076c191a082f7da108c924eae06842e1e49533

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ansi-html:0.0.7

Details: This affects all versions of package ansi-html. If an attacker provides a malicious string, it will get stuck processing the input for an extremely long time.

Type: Vulnerability

ReferenceHash: 8a670771bea65e399ea47b231dc302a6e080e35f525e45fb7b75a89ac9aa2ec7

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: glob-parent:5.1.0

Details: The software does not properly restrict the size or amount of resources that are requested or influenced by an actor, which can be used to consume more resources than intended.

Type: Vulnerability

ReferenceHash: 05059fc1fcd33dc8fcad398e12b3ba7a7a47e56e4b85f51eab8610c21da6a228

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ini:1.3.5

Details: The software does not properly protect an assumed-immutable element from being modified by an attacker.

Type: Vulnerability

ReferenceHash: 548d347262a3212f00625103f2f93f252472df0aa07d6b3168f52631b596eee9

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: kind-of:3.2.2

Details: Versions of `kind-of` 6.x prior to 6.0.3 are vulnerable to a Validation Bypass. A maliciously crafted object can alter the result of the type check, allowing attackers to bypass the type checking validation.

Recommendation

Upgrade to versions 6.0.3 or later.

Type: Vulnerability

ReferenceHash: 6b91c39506694e1521f9238dddfd22d23aa99ed960512d285cd4fa735a72da88

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: minimist:0.0.10

Details: The software constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment.

Type: Vulnerability

ReferenceHash: 259f9dbf2153775362b5c0eb9bb04160672d208d14c94aad677a27a8b6ae7580

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: node-forge:0.9.0

Details: The product does not validate or incorrectly validates input that can affect the control flow or data flow of a program.

Type: Vulnerability

ReferenceHash: 55785328fe937c02b3ad055b11ca917d926a8ae99791c82cbad12def8576c634

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: postcss:7.0.17

Details: The software does not properly restrict the size or amount of resources that are requested or influenced by an actor, which can be used to consume more resources than intended.

Type: Vulnerability

ReferenceHash: 53151e4a6fa035784946727 81e42320e0368074fcfb6f0014a2ffb85c94c0629

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: socket.io-parser:3.2.0

Details: The `socket.io-parser` npm package before versions 3.3.2 and 3.4.1 allows attackers to cause a denial of service (memory consumption) via a large packet because a concatenation approach is used.

Type: Vulnerability

ReferenceHash: 8332aec28800276ee5c41e49f5d13d9fdae65a04ac1e63838ace5c6dc9cbfbe4

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tar:4.4.13

Details: The software uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the software does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.

Type: Vulnerability

ReferenceHash: 16505940daa6e64554fcc322fa960bf4734571b6a6261a8d2865ccd0f6cec486

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: tree-kill:1.2.1

Details: The software constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

Type: Vulnerability

ReferenceHash: 5c2e94c1b1ec51f5bb93c1d448f0b07e0437a58b16240eac97e6a630b2e925ef

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: trim-newlines:1.0.0

Details: The trim-newlines package before 3.0.1 and 4.x before 4.0.1 for Node.js has an issue related to regular expression denial-of-service (ReDoS) for the .end() method.

Type: Vulnerability

ReferenceHash: f3c50be55f9830a653abdce9b41c88e2ebf334be9f1cf0ba51ce178bcf9a7597

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: xmlhttprequest-ssl:1.5.5

Details: The software does not validate, or incorrectly validates, a certificate.

Type: Vulnerability

ReferenceHash: e8a5a33773736a8e4c5e814e8fe81e6a4a29da3eef6196d4bd451c3db740b45a

==========

Language: Generic

Severity: HIGH

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: yargs-parser:11.1.1

Details: The software does not properly restrict the size or amount of resources that are requested or influenced by an actor, which can be used to consume more resources than intended.

Type: Vulnerability

ReferenceHash: e6e2d5547f989ef812ceac38cb86c424d4d66b3b19b0564e6b3bc19ed1159fb2

==========

Language: Generic

Severity: MEDIUM

Line: 0

Column: 0

SecurityTool: Trivy

Confidence: MEDIUM

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: minimist

Details: minimist before 1.2.2 could be tricked into adding or modifying properties of Object.prototype using a "constructor" or "__proto__" payload.

Installed Version: "0.0.8", Update to Version: "1.2.3, 0.2.1" for fix this issue.

PrimaryURL: https://avd.aquasec.com/nvd/cve-2020-7598.

Cwe Links: (https://cwe.mitre.org/data/definitions/20.html)

Type: Vulnerability

ReferenceHash: 08469a1478932e407905d1562550ad874d785b992b34ec8ef8ae0b4b72525104

==========

Language: JavaScript

Severity: MEDIUM

Line: 465

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ansi-regex

Details: ansi-regex is vulnerable to Inefficient Regular Expression Complexity

Type: Vulnerability

ReferenceHash: a1128ac5902a00377c619b6d2f87a8c9b5f67cd54a67d8f36cb2e839ed14ba37

==========

Language: JavaScript

Severity: MEDIUM

Line: 2872

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: browserslist

Details: The package browserslist from 4.0.0 and before 4.16.5 are vulnerable to Regular Expression Denial of Service (ReDoS) during parsing of queries.

Type: Vulnerability

ReferenceHash: 0d30b9bf12036929feb8c7f01b27bee39ff2a8ec748eff57379b4e9f650bc140

==========

Language: JavaScript

Severity: MEDIUM

Line: 12847

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: yargs-parser

Details: Affected versions of `yargs-parser` are vulnerable to prototype pollution. Arguments are not properly sanitized, allowing an attacker to modify the prototype of `Object`, causing the addition or modification of an existing property that will exist on all objects.

Parsing the argument `--foo.__proto__.bar baz'` adds a `bar` property with value `baz` to all objects. This is only exploitable if attackers have control over the arguments being passed to `yargs-parser`.

Recommendation

Upgrade to versions 13.1.2, 15.0.1, 18.1.1 or later.

Type: Vulnerability

ReferenceHash: 48dafd4c1ccf1e08bed319067d7ccb1437eb5e82cb0213281d36392a8ba3138b

==========

Language: JavaScript

Severity: MEDIUM

Line: 10948

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: sockjs

Details: Incorrect handling of Upgrade header with the value websocket leads in crashing of containers hosting sockjs apps. This affects the package sockjs before 0.3.20.

Type: Vulnerability

ReferenceHash: c2ffe8192d09439a0baa89b16f427f2893678735c60f137565973fbee2fbe616

==========

Language: JavaScript

Severity: MEDIUM

Line: 5547

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: minimist

Details: Affected versions of `minimist` are vulnerable to prototype pollution. Arguments are not properly sanitized, allowing an attacker to modify the prototype of `Object`, causing the addition or modification of an existing property that will exist on all objects.

Parsing the argument `--__proto__.y=Polluted` adds a `y` property with value `Polluted` to all objects. The argument `--__proto__=Polluted` raises and uncaught error and crashes the application.

This is exploitable if attackers have control over the arguments being passed to `minimist`.

Recommendation

Upgrade to versions 0.2.1, 1.2.3 or later.

Type: Vulnerability

ReferenceHash: 42548026ef828fd07914f31fd172720ebd6bf37343736f7c41f94e0cec3c40a0

==========

Language: JavaScript

Severity: MEDIUM

Line: 9343

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: postcss

Details: The npm package `postcss` from 7.0.0 and before versions 7.0.36 and 8.2.10 is vulnerable to Regular Expression Denial of Service (ReDoS) during source map parsing.

Type: Vulnerability

ReferenceHash: b3ef3fbd91c517952e3323469fba49db01636e7320b903d6a567b8d002ac3de8

==========

Language: JavaScript

Severity: MEDIUM

Line: 12607

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ws

Details: ### Impact

A specially crafted value of the `Sec-Websocket-Protocol` header can be used to significantly slow down a ws server.

Proof of concept

```js
for (const length of [1000, 2000, 4000, 8000, 16000, 32000]) {
  const value = 'b' + ' '.repeat(length) + 'x';
```

```
  const start = process.hrtime.bigint();

  value.trim().split(/ *, */);

  const end = process.hrtime.bigint();

  console.log('length = %d, time = %f ns', length, end - start);
}
```

Patches

The vulnerability was fixed in ws@7.4.6 (https://github.com/websockets/ws/commit/00c425ec77993773d823f018f64a5c44e17023ff) and backported to ws@6.2.2 (https://github.com/websockets/ws/commit/78c676d2a1acefbc05292e9f7ea0a9457704bf1b) and ws@5.2.3 (https://github.com/websockets/ws/commit/76d47c1479002022a3e4357b3c9f0e23a68d4cd2).

Workarounds

In vulnerable versions of ws, the issue can be mitigated by reducing the maximum allowed length of the request headers using the [`--max-http-header-size=size`](https://nodejs.org/api/cli.html#cli_max_http_header_size_size) and/or the [`maxHeaderSize`](https://nodejs.org/api/http.html#http_http_createserver_options_requestlistener) options.

Credits

The vulnerability was responsibly disclosed along with a fix in private by [Robert McLaughlin](https://github.com/robmcl4) from University of California, Santa Barbara.

Type: Vulnerability

ReferenceHash: fb5bc7a895ea795890ecc6c1575a42bcd91e55f8cd1cfb4972caf8909a3bc62a

==========

Language: JavaScript

Severity: MEDIUM

Line: 10826

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: socket.io

Details: The package socket.io before 2.4.0 are vulnerable to Insecure Defaults due to CORS Misconfiguration. All domains are whitelisted by default.

Type: Vulnerability

ReferenceHash: 85855ad4156ccaae0ecedc8ce40501457c4d11cd0cab162c37d6897480945e6f

==========

Language: JavaScript

Severity: MEDIUM

Line: 5374

Column:

SecurityTool: NpmAudit

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: minimist

Details: Affected versions of `minimist` are vulnerable to prototype pollution. Arguments are not properly sanitized, allowing an attacker to modify the prototype of `Object`, causing the addition or modification of an existing property that will exist on all objects.

Parsing the argument `--__proto__.y=Polluted` adds a `y` property with value `Polluted` to all objects. The argument `--__proto__=Polluted` raises and uncaught error and crashes the application.

This is exploitable if attackers have control over the arguments being passed to `minimist`.

Recommendation

Upgrade to versions 0.2.1, 1.2.3 or later.

Type: Vulnerability

ReferenceHash: 6e9321e7d1b6aefdee7d1dd8ed75426c6c538d857e1387ddd9749e474faebf4a

==========

Language: Generic

Severity: MEDIUM

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ansi-regex:3.0.0

Details: ansi-regex is vulnerable to Inefficient Regular Expression Complexity

Type: Vulnerability

ReferenceHash: 40af99e45ee5cce71ab89b01e109153bdab2965c7fd5346001a7dff71bb8a6f2

==========

Language: Generic

Severity: MEDIUM

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: browserslist:4.7.3

Details: The package browserslist from 4.0.0 and before 4.16.5 are vulnerable to Regular Expression Denial of Service (ReDoS) during parsing of queries.

Type: Vulnerability

ReferenceHash: c7b81c26feab0de966b6003991f84e1206d0ca62e6974ec4b8b51ce09503503b

==========

Language: Generic

Severity: MEDIUM

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: socket.io:2.1.1

Details: The software does not properly verify that the source of data or communication is valid.

Type: Vulnerability

ReferenceHash: 9997a39cf84d9fd7beff3fd737a3e2fb6d3128d480d8834cd04a1bc3f01141d0

==========

Language: Generic

Severity: MEDIUM

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: sockjs:0.3.19

Details: Incorrect handling of Upgrade header with the value websocket leads in crashing of containers hosting sockjs apps. This affects the package sockjs before 0.3.20.

Type: Vulnerability

ReferenceHash: 72e6f00d39295a215776de2dc3307724d505848b6618548fef39f5b5df76af73

==========

Language: Generic

Severity: MEDIUM

Line:

Column:

SecurityTool: OwaspDependencyCheck

Confidence: LOW

File: /home/andrey/github/innereye/web/ui/package-lock.json

Code: ws:3.3.3

Details: ### Impact

A specially crafted value of the `Sec-Websocket-Protocol` header can be used to significantly slow down a ws server.

Proof of concept

```js
for (const length of [1000, 2000, 4000, 8000, 16000, 32000]) {
  const value = 'b' + ' '.repeat(length) + 'x';
  const start = process.hrtime.bigint();
  value.trim().split(/ *, */);
  const end = process.hrtime.bigint();
  console.log('length = %d, time = %f ns', length, end - start);
}
```

Patches

The vulnerability was fixed in ws@7.4.6 (https://github.com/websockets/ws/commit/00c425ec77993773d823f018f64a5c44e17023ff) and backported to ws@6.2.2

(https://github.com/websockets/ws/commit/78c676d2a1acefbc05292e9f7ea0a9457704bf1b) and ws@5.2.3 (https://github.com/websockets/ws/commit/76d47c1479002022a3e4357b3c9f0e23a68d4cd2).

Workarounds

In vulnerable versions of ws, the issue can be mitigated by reducing the maximum allowed length of the request headers using the [`--max-http-header-size=size`](https://nodejs.org/api/cli.html#cli_max_http_header_size_size) and/or the [`maxHeaderSize`](https://nodejs.org/api/http.html#http_http_createserver_options_requestlistener) options.

Credits

The vulnerability was responsibly disclosed along with a fix in private by [Robert McLaughlin](https://github.com/robmcl4) from University of California, Santa Barbara.

Type: Vulnerability

ReferenceHash: 94c82f8c1389b707e06e10b16e5393732edca8c3e4256d9932516b352a9fb9bb

==========

In this analysis, a total of 47 possible vulnerabilities were found and we classified them into:

Total of Vulnerability CRITICAL is: 2

Total of Vulnerability HIGH is: 30

Total of Vulnerability MEDIUM is: 15

==========