**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кваліфікаційна наукова
праця на правах рукопису

**СУСУКАЙЛО ВІТАЛІЙ АНДРІЙОВИЧ**

УДК 004.056.53

**ДИСЕРТАЦІЯ**

# РОЗРОБЛЕННЯ МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ ДЛЯ СКЛАДОВИХ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ

125 Кібербезпека

12 “Інформаційні технології”

Подається на здобуття наукового ступеня доктора філософії
Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело _____ /Сусукайло Віталій Андрійович/

Науковий керівник: Опірський Іван Романович, доктор технічних наук, професор

Львів – 2024

# АНОТАЦІЯ

*Сусукайло В.А.* **Розроблення моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем.** – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю “125 – Кібербезпека”. – Національний університет “Львівська політехніка” Міністерства освіти і науки України, Львів, 2024.

Своєчасне розслідування кіберзлочинів стає важливим завданням для працівників сфери кібербезпеки, зважаючи на розвиток та вплив інформаційних технологій. Оскільки компанії, уряди та окремі особи переводять переважну більшість своїх операцій і особистої діяльності в Інтернет та використовують хмарні рішення, вразливість цифрового простору стає різко очевидною. Будь-яке зволікання з розслідуванням кіберзлочинів може призвести не тільки до значних фінансових втрат, але й до компрометації конфіденційних даних, підірвати довіру громадськості та навіть поставити під загрозу національну безпеку у разі атак на критичну інфраструктуру. У даній роботі автор зосереджується на розробці моделі системи дослідження кіберзлочинів, у якій використано традиційні принципи аналізу інцидентів інформаційної безпеки та запропоновано модель штучного інтелекту.

Під системою дослідження кіберзлочинів автор має на увазі комплексний механізм, який інтегрує функції дослідження вразливостей, збір даних про загрози кібербезпеці, засоби для ідентифікації кібератаки та аналізу кіберзлочину, як факту успішної атаки. На відміну від сучасних систем типу IDS/IPS (системи виявлення та запобігання вторгненням) або брандмауерів (міжмережевих екранів) дані системи є більш комплексними, оскільки вони не лише виявляють зловмисні дії, але й здійснюють глибокий аналіз поведінки користувачів, мережевого трафіка та додатків, використовуючи розширені алгоритми машинного навчання та штучного інтелекту.

В історичному плані такі комплексні системи дослідження кіберзлочинів не могли бути створені раніше через обмежені технологічні можливості, відсутність

необхідних обсягів даних для аналізу та недостатньо розвинені методології штучного інтелекту.

Теперішній стан світової кібербезпеки та стрімкий розвиток методів та засобів захисту дозволяють створювати гібридні системи, які використовують переваги різних підходів і технологій. Такі системи дослідження кіберзлочинів є втіленням найсучасніших досягнень у галузі кібербезпеки, вони забезпечують більш глибоке розуміння загроз і ефективнішу протидію кіберзлочинності.

Дослідження включає аналіз наявних проблем комплексних систем дослідження кіберзлочинів для інфраструктури інформаційних систем, вивчення сучасних методів побудови систем інформаційної безпеки з використанням DevSecOps-підходу, ISMS-методології, технології Blockchain та моделей штучного інтелекту. А також розробку моделі дослідження кіберзлочинів для складових інфраструктури інформаційних систем на різних її рівнях на основі штучного інтелекту, для підвищення ефективності виявлення та опрацювання першопричин вторгнень та її усунення.

Об’єктом дослідження є система дослідження кіберзлочинів з використанням штучного інтелекту та підходу DevSecOps, що дозволяє виявляти першопричини вторгнення та визначення вразливих елементів інформаційної системи з метою покращення ефективності її захисту та забезпечення системи відповідності міжнародним стандартам інформаційної безпеки .

Предметом дослідження є методи та засоби виявлення та дослідження кіберзлочинів в інформаційних системах та їх компонентах з метою підвищення ефективності захищеності від кібератак.

У процесі досліджень використано методи оптимізації, імітаційного та аналітичного моделювання, математичної статистики, об'єктно-орієнтованого програмування, теорії інформації та кодування.

У першому розділі **"Аналіз стану проблеми розроблення комплексної системи дослідження кіберзагроз інфраструктури інформаційних систем"** проводиться аналіз компонентів, які складають інфраструктуру сучасної інформаційної системи, аналіз сучасних тенденцій та технологій, щоб забезпечити

повне розуміння поточних стандартів інфраструктури інформаційних систем. Також у цьому розділі проаналізовано перехід від традиційної апаратної інфраструктури до хмарних рішень. Зіставлено обидва підходи, окреслено переваги, недоліки та можливості масштабованості. Це порівняння допомагає оцінити ризики кожного з підходів. Також у розділі звернено увагу на актуальні проблеми дослідження кіберзлочинів та проаналізовано законодавчі акти України, пов'язані з кіберзлочинністю, для оцінки проблем дослідження кіберзлочинів в Україні.

У другому розділі **"Дослідження використання властивостей моделей штучного інтелекту для виявлення кібератак та аналізу кіберзлочинів"** проаналізовано основні компоненти систем моніторингу інформаційної безпеки, дані з систем розвідки про загрози інформаційної безпеки та зосереджено увагу на аналізі використання ШІ та сучасних моделей систем дослідження загроз інформаційній безпеці для компонентів інфраструктури інформаційних систем. У даному розділі описано комплексне дослідження, зосереджене навколо аналізу та оптимізації систем дослідження кіберзлочинів. Оскільки цифровий ландшафт стає все складнішим, важливість вдосконаленої системи аналізу загроз неможливо недооцінити, тому автором було проаналізовано рішення, що використовується для виявлення кібератак та дослідження кіберзлочинів. Також автор проаналізував використання алгоритмів машинного навчання для дослідження аномалій у журналах подій та використання алгоритмів GPT для аналізу кібератак. Це гарантує, що дане дослідження фокусується на сучасних системах дослідження кіберзлочинів та обґрунтуванні припущення, що моделі штучного інтелекту можуть бути використані для виявлення та аналізу кіберзлочинів, що було підтверджено описаними у цьому розділі експериментами.

Третій розділ **"Розробка моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем"** присвячений розробці методології, моделі та архітектури комплексної системи дослідження кіберзлочинів. У цьому розділі детально проаналізовано основні компоненти моделі системи дослідження кіберзлочинів, зокрема елементи DevSecOps-процесу,

інтегрованого в модель комплексної системи дослідження кіберзлочинів, запропоновано системи дослідження загроз та систему управління вразливостями як складові системи дослідження кіберзлочинів. У межах цього розділу представлено роль приманок на основі Blockchain-технології для дослідження та даних отриманих з приманок. Модель демонструє, як інтеграція підходу DevSecOps, приманок на основі Blockchain-технології та елементів штучного інтелекту можуть бути використані для дослідження кіберзлочинів елементів інфраструктури інформаційних систем. Крім того, представлено архітектуру комплексної системи дослідження.

У четвертому розділі **"Дослідження ефективності моделі системи дослідження кіберзлочинів"** проведено дослідження можливостей аналізу кіберзлочинів, використовуючи запропоновану систему дослідження кіберзлочинів, проведено низку експериментів та задокументовано висновки з рівня безпеки та оцінка ефективності системи в порівнянні з існуючими рішеннями.

Загалом з’ясовано, що розроблений прототип системи є більш швидким при виявленні кібератак та аналізі кіберзлочинів , ніж традиційні аналоги.

Подальша оптимізація прототипу системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем дозволить покращити результат та вдосконалити систему.

У **висновках** дисертаційної роботи викладено основні результати і рекомендації, які випливають з проведених досліджень, представлено та охарактеризовано кількісні оцінки показників ефективності в умовах використання запропонованих рішень.

У **додатках** до дисертації долучено програмні коди реалізації імітаційних моделей, акти впровадження результатів дисертаційної роботи, а також список наукових праць і апробацій автора за темою дисертації.

**Ключові слова:** кібербезпека, кіберзлочин, інцидент безпеки, виявлення аномалій, масив даних, штучний інтелект, інфраструктура, інформаційні системи, Blockchain, ізоляційний ліс, DevSecOps, GPT, система управління інформаційною безпекою, ISO 27001.

## ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

**Наукові праці, в яких опубліковано наукові результати дисертації:**

1. Опірський І. Р., Василишин С. І., Сусукайло В. А. Аналіз загроз та безпеки технології NFC при передачі даних для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). – С. 37–44.

2. Опірський І. Р., Сусукайло В. А., Василишин С. І., Луковський Т. І. Розробка методу використання технології NFC для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). – С. 151–157.

3. Опірський І.Р., Василишин С.І., Сусукайло В.А. Розслідування кіберзлочинів за допомогою приманок у хмарному середовищі // Безпека інформації. –2021. – 27(1). – С.13–20. https://doi.org/10.18372/2225-5036.26.15574

4. Vasylyshyn S., Susukailo V., Opirskyy I., Kurii Y., Tyshyk I. A model of decoy system based on dynamic attributes for cybercrime investigation // Eastern-European Journal of Enterprise Technologies. – 2023.– 1 (9 (121)), pp. 6–20. https://doi.org/10.15587/1729-4061.2023.273363 (Scopus)

5. Сусукайло В. Використання підходу DevSecOps для аналізу сучасних загроз інформаційної безпеки // Кібербезпека: освіта, наука, техніка. – 2021. – Вип. 2, вип. 14. – С. 26–35.

6. Опірський І. Р., Сусукайло В. А., Василишин С. І. Дослідження можливостей використання чатботів зі штучним інтелектом для дослідження журналів подій // Захист інформації. – 2022. – Т. 24, № 4. – С. 177–183.

7. Kostiak M., Yevseiev S., Pohasii S., Zhuchenko O., Milov O., Lysechko V., Kovalenko O., Volkov A., Lezik A., Susukailo V. Development of crypto-code constructs based on LDPC codes // Східно-Європейський журнал передових технологій. – 2022. – № 2/9 (116). – P. 44–59

8. Susukailo V., Opirskyy I., Yaremko O. Methodology of ISMS Establishment Against Modern Cybersecurity Threats // Lecture Notes in Electrical Engineering. – 2022.

– Vol. 831: Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks. – p. 257–271.

9. Сусукайло В. А., Опірський І. Р., Піскозуб А. З., Волошин Р. Я., Друзюк О. С. Аналіз атак, що використовуються кіберзлочинцями під час пандемії covid 19 // Захист інформації. – 2021. – Т. 22, № 4. – С. 220–226.

10. Опірський І. Р., Курій Є. О., Сусукайло В. А. Розробка методології оцінки відповідності стандарту ISO 27001 // Захист інформації. – 2023. – Т. 25, № 3. – С. 132–139.

***Наукові праці, які засвідчують апробацію матеріалів дисертації:***

11. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the use of software baits (honeypots) as a means of ensuring information security // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 242–245, 9321925, DOI: 10.1109/CSIT49958.2020.9321925 (Scopus)

12. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the attack vectors used by threat actors during pandemic // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 261–264, 9321897, DOI: 10.1109/CSIT49958.2020.9321897 (Scopus)

13. Опірський І.Р., Василишин С.І. Сусукайло В.А. Дослідження вразливості Zerologon // “Технічні засоби захисту інформації”, семінар при вченій раді НАН України, Київ, Україна, 2021.

14. Susukailo V., Opirskyy I., Vasilishyn S. Analysis of the possibility of using chatbots with Artificial Intelligence to detect information security incidents // Захист інформації і безпека інформаційних систем : матеріали IX Міжнародної науково-технічної конференції (Львів, 25–26 травня 2023 р.). – 2023. – С. 120–121.

15. Opirskyy I., Tyshyk I., Susukailo V. Evaluation of the possibility of Realizing the Crime of the Information System at Different Stages of TCP/IP // 2021 IEEE 4th International conference on advanced information and communication technologies: conference proceedings AICT-2021 (Lviv, Ukraine, September 21-25, 2021). – 2021. – С. 261–265.

16. Susukailo V., Vasilishyn S., Opirskyy I., Buriachok V., Riabchun O. Cybercrimes investigation via honeypots in cloud environments // CEUR Workshop Proceedings. – 2021. – Vol. 2923: Proceedings of selected papers of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021), Kyiv, Ukraine, January 28, 2021 (online). – p. 91–96.

17. Сусукайло В. А., Опірський І. Р. Дослідження можливостей системи Azure Log Analytics для аналізу інцидентів інформаційної безпеки в хмарних рішеннях // Інформаційна безпека та інформаційні технології : збірник тез доповідей IV Всеукраїнської науково-практичної конференції молодих учених, студентів та курсантів (Львів, 27 листопада 2020). – 2020. – С. 57–59.

18. Susukailo V., Opirskyy I., Kret T. Advantages of Threat Hunting with Endpoint Detection and Response Solutions // Захист Інформації і Безпека Інформаційних систем: VII Міжнародна науково-технічна конференція “Захист інформації та безпека інформаційних систем” – 2019. – С. 17–19.

## SUMMARY

*Susukailo V.A.* **DEVELOPMENT OF A CYBERCRIME INVESTIGATION SYSTEM MODEL FOR INFORMATION SYSTEM INFRASTRUCTURE COMPONENTS.** – Qualifying scientific work on the rights of the manuscript.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in the specialty 125 " Cyber Security". - Lviv Polytechnic National University, Lviv, 2024.

*Abstract content.* Timely investigation of cybercrimes becomes an essential task for employees in the field of cyber security, considering the development and impact of information technologies. As companies, governments, and individuals move most of their operations and personal activities online and use cloud solutions, the vulnerabilities of the digital space are becoming starkly apparent. Any delay in the investigation of cybercrimes can lead not only to significant financial losses but also to the compromise of sensitive data, undermine public trust, and even jeopardize national security in the event of attacks on critical infrastructure. In this work, the author focuses on developing a cybercrime investigation system model that uses traditional information security incident analysis principles and proposes an artificial intelligence model.

Under the system of cybercrime investigation, the author means a complex mechanism that integrates the functions of vulnerability research, data collection on cyber security threats, tools for identifying a cyberattack, and analysis of cybercrime as a fact of a successful attack. Unlike modern systems such as IDS/IPS (intrusion detection and prevention systems) or firewalls (internet screens), these systems are more comprehensive, as they detect not only malicious actions but also perform deep analysis of user behavior, network traffic, and applications using advanced algorithms of machine learning and artificial intelligence.

Historically, such complex cybercrime investigation systems could not have been created earlier due to limited technological capabilities, a lack of the necessary data volumes for analysis, and insufficiently developed artificial intelligence methodologies.

The current state of global cyber security and the rapid development of methods and means of protection make it possible to create hybrid systems that take advantage of different approaches and technologies. Such cybercrime investigation systems embody

the most modern achievements in cyber security; they provide a deeper understanding of threats and more effective countermeasures against cybercrime.

The study includes the analysis of existing problems of complex cybercrime investigation systems for the infrastructure of information systems, the study of modern methods of building information security systems using the DevSecOps approach, ISMS methodology, Blockchain technology, and artificial intelligence models as well as the development of a cybercrime investigation model for the components of the infrastructure of information systems at its various levels based on artificial intelligence, to increase the effectiveness of identifying and working out the root causes of intrusions and their elimination.

The study's object is a cybercrime investigation system using artificial intelligence and the DevSecOps approach. This system allows for identifying the root causes of intrusion and vulnerable elements of an information system to improve its protection's effectiveness and ensure the system's compliance with international information security standards.

The study's subject is methods and means of detecting and investigating cybercrimes in information systems and their components to improve the effectiveness of protection against cyberattacks.

Optimization methods, simulation, analytical modeling, mathematical statistics, object-oriented programming, information theory, and coding were used in the research process.

In the first chapter, "Analysis of the state of the problem of developing a comprehensive system for researching cyber threats to the infrastructure of information systems," an analysis of the components that make up the infrastructure of a modern information system, an analysis of modern trends and technologies is carried out to ensure a complete understanding of the current standards of the infrastructure of information systems. This section also analyzes the transition from traditional hardware infrastructure to cloud solutions. Both approaches are compared, and advantages, disadvantages, and scalability possibilities are outlined. This comparison helps assess the risks of each approach. The chapter also draws attention to the current problems of

cybercrime investigation. It analyzes Ukraine's legislative acts related to cybercrime to assess the problems of cybercrime investigation in Ukraine.

In the second chapter, "Research on the use of the properties of artificial intelligence models for the detection of cyberattacks and the analysis of cybercrimes," the main components of information security monitoring systems and data from intelligence systems on information security threats are analyzed. Attention is focused on analyzing the use of AI and modern models of information security threat research systems for components infrastructure of information systems. This chapter describes a comprehensive study on analyzing and optimizing cybercrime investigation systems. As the digital landscape becomes increasingly complex, the importance of an advanced threat analysis system cannot be understated, so the author analyzed the solution used to detect cyberattacks and investigate cybercrimes. The author also analyzed the machine learning algorithms used to investigate anomalies in event logs and GPT algorithms for cyber attack analysis. This ensures that this research focuses on modern cybercrime investigation systems and substantiation the assumption that artificial intelligence models can be used to detect and analyze cybercrimes, which has been confirmed by the experiments described in this section.

The third section, "Development of a cybercrime investigation system model for information system infrastructure components," is devoted to developing the methodology, model, and architecture of a comprehensive cybercrime investigation system. In this section, the main components of the cybercrime investigation system model are analyzed in detail; in particular, the elements of the DevSecOps process integrated into the model of the comprehensive cybercrime investigation system, the threat investigation system, and the vulnerability management system are proposed as components of the cybercrime investigation system. This section presents the role of blockchain-based decoys for research and decoy-derived data. The model demonstrates how the integration of the DevSecOps approach, decoys based on Blockchain technology, and elements of artificial intelligence can be used to investigate cybercrimes of elements of the infrastructure of information systems. In addition, the architecture of the complex research system is presented.

In the fourth chapter, "Analysis of the effectiveness of the cybercrime investigation system model" a study of the possibilities of analyzing cybercrimes using the proposed cybercrime investigation system is carried out, and several experiments are conducted. Conclusions from the level of security are documented. The system's effectiveness is evaluated in comparison with existing solutions.

Generally, the developed system prototype is faster than traditional analogs in detecting cyberattacks and analyzing cybercrimes.

Further optimization of the cybercrime investigation system prototype for the information systems infrastructure components will allow for improved results and the system.

**Keywords:** cybercrime, cyberattack, security incident, anomaly detection, dataset artificial intelligence, infrastructure, information system, Blockchain, isolation forest, DevSecOps, GPT,information security management system, ISO 27001.

## LIST OF PUBLICATIONS OF THE ACQUIRER

**Scientific works in which the main scientific results of the dissertation are published:**

1. Opirskyy I.R., Vasylyshyn S.I., Susukailo V.A. Analysis of threats and security of NFC technology during data transmission for automated replication of the user profile // Bulletin of the Eastern Ukrainian National University named after Volodymyr Dahl. Informational security. – 2018. – No. 3/4 (31/32). – p. 37–44.

2. Opirskyy I.R., Susukailo V.A., Vasylyshyn S.I., Lukovskyi T.I. Development of a method of using NFC technology for automated user profile replication // Bulletin of the East Ukrainian National University named after Volodymyr Dahl. Informational security. – 2018. – No. 3/4 (31/32). - Pp. 151–157.

3. Opirskyy I.R., Vasylyshyn S.I., Susukailo V.A. Withered Investigating cybercrimes using decoys in the cloud. Information Security, 27(1). – 2021. – C.13–20. https://doi.org/10.18372/2225-5036.26.15574

4. Vasylyshyn S., Susukailo V., Opirskyy I., Kurii Y., Tyshyk I. A model of decoy system based on dynamic attributes for cybercrime investigation // Eastern-European Journal of Enterprise Technologies. – 2023. – 1 (9 (121)), pp. 6–20. https://doi.org/10.15587/1729-4061.2023.273363 (Scopus)

5. Susukailo V. Using the DevSecOps approach to analyze modern information security threats // Cybersecurity: Education, Science, Technology. – 2021. – vol. 2, issue 14, pp. 26–35.

6. Opirskyy I.R., Susukailo V.A., Vasylyshyn S.I. Study of the possibilities of using chatbots with artificial intelligence for the study of event logs // Information Protection. – 2022. – Vol. 24, No. 4. – P. 177–183.

7. Kostiak M., Yevseiev S., Pohasii S., Zhuchenko O., Milov O., Lysechko V., Kovalenko O., Volkov A., Lezik A., Susukailo V. Development of crypto-code constructs based on LDPC codes // Eastern European Journal of Advanced Technologies. – 2022. – No. 2/9 (116). – Pp. 44–59

8. Susukailo V., Opirskyy I., Yaremko O. Methodology of ISMS Establishment Against Modern Cybersecurity Threats // Lecture Notes in Electrical Engineering. – 2022.

– Vol. 831: Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks. – Pp. 257–271.

9. Susukailo V.A., Opirskyy I.R., Piskozub A.Z., Voloshyn R.Ya., Druzyuk O.S. Analysis of attacks used by cybercriminals during the covid 19 pandemic // Protection of information. – 2021. – Vol. 22, No. 4. – P. 220–226.

10. Opirskyy I.R., Kurii E.O., Susukaylo V.A. Development of methodology for assessing compliance with the ISO 27001 standard // Protection of information. – 2023. – Vol. 25, No. 3. – Pp. 132–139.

**Scientific works certifying the approval of the dissertation materials:**

11. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the use of software baits (honeypots) as a means of ensuring information security // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 242–245, 9321925, DOI: 10.1109/CSIT49958.2020.9321925 (Scopus)

12. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the attack vectors used by threat actors during the pandemic // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 261–264, 9321897, DOI: 10.1109/CSIT49958.2020.9321897 (Scopus)

13. Opirskyy I.R., Vasylyshyn S.I., Susukailo V.A., Zerologon Vulnerability Study // "Technical means of information protection", seminar at the Scientific Council of the National Academy of Sciences of Ukraine, Kyiv, Ukraine, 2021.

14. Susukailo V., Opirskyy I., Vasylyshyn S. Analysis of the possibility of using chatbots with Artificial Intelligence to detect information security incidents // Protection of information and security of information systems: materials of the IX International Scientific and Technical Conference (Lviv, 25–26 May 2023). – 2023. – C. 120–121

15. Opirskyy I., Tyshyk I., Susukailo V. Evaluation of the possibility of Realizing the Crime of the Information System at Different Stages of TCP/IP // 2021 IEEE 4th International conference on advanced information and communication technologies: conference proceedings AICT- 2021 (Lviv, Ukraine, September 21-25, 2021). – 2021. – C. 261–265.

16. Susukailo V., Vasylyshyn S., Opirskyy I., Buriachok V., Riabchun O. Cybercrimes investigation via honeypots in cloud environments // CEUR Workshop Proceedings. – 2021. – Vol. 2923: Proceedings of selected papers of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021), Kyiv, Ukraine, January 28, 2021 (online). – Pp. 91–96.

17. Susukailo V. A., Opirskyy I. R. Researching the possibilities of the Azure Log Analytics system for the analysis of information security incidents in cloud solutions // Information security and information technologies: a collection of abstracts of reports of the IV All-Ukrainian scientific and practical conference of young scientists, students and cadets (Lviv, November 27, 2020). – 2020. – Pp. 57–59.

18. Susukailo V., Opirskyy I., Kret T. Advantages of Threat Hunting with Endpoint Detection and Response Solutions // Information Protection and Security of Information Systems: VII International Scientific and Technical Conference "Information Protection and Security of Information Systems". – 2019. – Pp. 17-19.

**ЗМІСТ**

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IDS – Intrusion Detection System, система виявлення вторгнень.

IPS – Intrusion Prevention System, система запобігання вторгненням.

SIEM – Security Information and Event Management, система управління інформаційною безпекою та управління подіями безпеки.

DevSecOps – це практика інтеграції тестування безпеки в етапи процесу розробки програмного забезпечення.

EDR – Endpoint Detection and Response, це захисна система, яка концентрує свою увагу на кінцевих точках мережі та дотичних елементів.

SAST – Static Application Security Testing, статичне тестування безпеки додатків.

DAST – Dynamic Application Security Testing, динамічне тестування безпеки додатків.

SCA – Software Composition Analysis, аналіз складу програмного забезпечення.

REST – Representational State Transfer, передача репрезентативного стану.

AI – Artificial intelligence, штучний інтелект.

IAM – Identity Access Management, управління обліковими даними.

SaaS – Software as a service, програмне забезпечення як сервіс.

PaaS – Platform as a Service, платформа як сервіс.

IaaS – Infrastructure as a Service, інфраструктура як сервіс.

CI – Continuous Integration, безперервна інтеграція.

CD –Continuous Delivery, безперервне впровадження.

NIDS – Network Intrusion Detection System, система виявлення вторгнень у мережі.

ISMS – Information Security Management System, система управління інформаційною безпекою.

# ВСТУП

**Актуальність.** Ландшафт кіберзлочинності та кібербезпеки у період між 2020 та 2023 роками демонструє переконливу статистику, зокрема щодо часу, необхідного для виявлення загроз, і переважаючих тенденцій у кіберзагрозах. Згідно з дослідженням, проведеним Qualys, 25% поширених вразливостей і вразливостей (CVE) з високим ризиком були використані в той самий день, коли вони були оприлюднені. Крім того, протягом приблизно трьох тижнів (19 днів) після публікації 75% цих вразливостей було використано, що підкреслює критичне вікно для організацій, щоб усунути ці вразливості [1]. Згідно зі статистикою компанії Varonis, програмні застосунки якої гарантують безпеку даних з точки зору шкідливого програмного забезпечення, 94% доставляється електронною поштою, що підкреслює важливість надійних протоколів безпеки електронної пошти. Крім того, повідомляється, що хакери атакують у середньому 26 000 разів на день, що зривається на атаку кожні три секунди. Така частота атак демонструє велику кількість кіберзагроз сьогодні [2]. Ще один цікавий аспект визначений організацією Varonis – вплив віддаленої роботи на кібербезпеку. Зазначається, що організаціям із віддаленим персоналом потрібно на 58 днів більше часу, щоб виявити та усунути порушення, порівняно з офісними організаціями. Цей розширений часовий проміжок вказує на додаткові проблеми, пов’язані з віддаленим робочим середовищем для підтримки надійної кібербезпеки [2].

Загалом ці статистичні дані описують ландшафт кіберзагроз, що скоро розвивається, де швидкість і точність у виявленні загроз і реагуванні на них є вирішальними.

Також важливо вчасно виявляти кібератаки, адже вони спрямовані на порушення цілісності, конфіденційності або доступності інформації, що часто призводить до кіберзлочинів, які за своєчасного виявлення можуть бути локалізовані. Інтеграція передових технологій, таких як ШІ, та адаптація до нових робочих середовищ є ключовими компонентами ефективної боротьби з кіберзлочинністю. Під системою дослідження кіберзлочинів автор має на увазі комплексну систему, яка інтегрує функції дослідження вразливостей, збору даних

про загрози та засоби для ідентифікації кібератаки та аналізу кіберзлочину, як факту успішної атаки на різних рівнях інфраструктури інформаційної системи. На відміну від систем виявлення та запобігання вторгненням або брандмауерів дані системи є більш комплексними, оскільки вони не лише виявляють зловмисні дії, але й здійснюють глибокий аналіз поведінки користувачів, мережевого трафіка та додатків, використовуючи розширені алгоритми машинного навчання та штучного інтелекту.

Використання штучного інтелекту (AI) в інформаційній безпеці є динамічною сферою, що розвивається. Для прикладу, дослідження Аріфа Алі Мугала, опубліковане в Journal of Artificial Intelligence and Machine Learning in Management підкреслює переваги штучного інтелекту в інформаційній безпеці, такі як його здатність швидко обробляти великі обсяги даних, виявляти аномалії та незвичні дії, автоматизувати реагування на загрози та надавати інформацію про події безпеки в реальному часі [3]. Зі створенням більших обсягів даних потреба в їх аналізі зростає. Управління великими компіляціями даних в асиметричній структурі даних стає все більшою проблемою для компаній. Подібним чином виклики на рівні операційних навичок, інтеграції даних та інформаційної технологічної інфраструктури вимагають постійного оновлення програмного забезпечення для керування даними для великих даних [4]. Одним із основних застосувань штучного інтелекту є аналіз даних. Алгоритми штучного інтелекту переглядають величезні набори даних, щоб виявити незвичайні моделі та аномалії. Це може включати виявлення порушень у мережевому трафіку, виявлення підозрілих електронних листів або виявлення незвичайних фінансових транзакцій. Розпізнаючи ці закономірності, штучний інтелект може попередити аналітиків безпеки про потенційні дії кіберзлочинців, які інакше могли б залишитися непоміченими.

Технології кібербезпеки покращують заходи безпеки для виявлення кібератак і реагування на них. Систем безпеки, які використовувались раніше, вже недостатньо, оскільки кіберзлочинці достатньо розумні, щоб уникнути звичайних систем безпеки. Звичайним системам безпеки не вистачає ефективності у виявленні раніше невидимих і поліморфних атак безпеки, тому методи машинного навчання

(ML) відіграють важливу роль у численних додатках кібербезпеки [5]. Машинне навчання займає перше місце у виявленні загроз у сфері кібербезпеки. Навчаючись на наборах даних про відомі кіберзагрози, ці моделі навчаються ідентифікувати та позначати подібні загрози в нових даних. Це надзвичайно важливо для виявлення різних форм зловмисного програмного забезпечення, включаючи віруси, програми-вимагачі та шпигунські програми, у режимі реального часу, що дозволяє швидше реагувати на ці загрози.

Є кілька типових робіт використання моделей ШІ для виявлення та дослідження кіберзлочинів. Проблема покращення системи дослідження кіберзлочинів та використання ШІ досліджувалась вченими, такими як: Шуай Чжоу, Майкл Горовіц, Джейкоб Сакіні, Роман Киричок та Роман Одарченко.та інші.

Незважаючи на велику кількість досліджень у даному напрямку, є низка невирішених проблеми, зокрема таких: комплексний підхід дослідження кіберзлочинів на різних рівнях інфраструктури інформаційної системи, забезпечення системи дослідження кіберзлочинів, що використовує алгоритми ШІ міжнародним стандартам, вплив підходу DevSecOps на дослідження кіберзлочинів.

Проведені експерименти з порівнянням моделей виявлення аномалій дають можливість створити прототип моделі системи дослідження кіберзлочинів, що може ефективно виявляти підозрілу діяльність та повідомляти аналітиків інформаційної безпеки для більш детального аналізу про атаки на різних рівнях інфраструктури інформаційної системи. Шляхом інтеграції систем аналізу загроз та систем дослідження вразливостей, що побудована з урахуванням підходу DevSecOps, аналітики інформаційної безпеки можуть комплексно приступати до аналізу кіберзлочинів та більш точно аналізувати події та інциденти інформаційної безпеки.

У цій дисертаційній роботі розглянено проблему аналізу кіберзлочинів, зокрема використання моделей штучного інтелекту для зменшення тривалості аналізу, не зменшуючи при цьому ефективність виявлення точно позитивних кібератак на різних рівнях інфраструктури інформаційної системи. Однак існують певні виклики та обмеження, які потрібно вирішувати, щоб забезпечити ефективне

дослідження.

Одним із викликів є комплексний підхід. Для детального дослідження кіберзлочинів потрібно забезпечити інтеграцію різних систем між собою. Систему дослідження загроз та систему управління вразливостями потрібно використати для отримання даних про інформаційну систему, на яку здійснюється атака.

Другий виклик полягає в забезпеченні відповідності системи міжнародним стандартам. Оскільки система дослідження кіберзлочинів повинна використовуватись як державними, так і приватними організаціями. Для можливості використання вона повинна відповідати міжнародним стандартам.

Дана дисертаційна робота описує всі переваги та недоліки дослідження кіберзлочинів шляхом використання моделей штучного інтелекту та підходу DevSecOps.

**Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційні дослідження виконано відповідно до наукового напряму кафедри захисту інформації Національного університету "Львівська політехніка".

Основні положення та результати дисертаційної роботи впроваджені у навчальний процес кафедри Захист інформації Національного університету "Львівська політехніка" з вивчення дисципліни "Безпека програмного забезпечення" для студентів напрямку підготовки "125 – Кібербезпека", освітньої програми "Управління інформаційною безпекою".

Дисертаційні дослідження виконано у відповідності до наукового напряму кафедри захисту інформації Національного університету "Львівська політехніка" - "Дослідження систем технічного захисту інформації, каналів зв'язку та комп'ютерних мереж, фізичного захисту інформації та криптографії", в межах кафедральної науково-дослідної роботи: "Розроблення та удосконалення методів і засобів захисту інформації для протидії несанкціонованому доступу в інформаційно-комунікаційних мережах" (шифр ЗІ-7) (№ держреєстрації 0119U101690) (2019р.-2022р.). Окремі частини роботи виконано в межах держбюджетної науково-дослідної роботи: «Дослідження стійкості біометричних

систем автентифікації до атак із застосуванням технології клонування голосу на основі глибинних нейронних мереж» (№ держреєстрації 0124U000407).

А також в діяльності підприємств ТОВ "Бінарікс Україна", ТОВ "ТЕХМЕДЖИК" та AG "Hiveon".

Мета роботи. Метою дисертаційної роботи є підвищення захищеності компонентів інформаційних систем від кібератак на різних рівнях її інфраструктури завдяки розробленню моделі системи дослідження кіберзлочинів з використанням штучного інтелекту та DevSecOps–підходу. Це дасть змогу підвищити захищеність інформаційних систем шляхом зменшення часу аналізу кіберзлочинів без втрат ефективності та забезпечити дослідження подій інформаційної безпеки, враховуючи вразливості інформаційної системи на різних її рівнях.

Завдання. Дисертаційна робота присвячена вирішенню актуального науково-практичного завдання з підвищення ефективності виявлення кіберзлочинів в інфраструктурі інформаційних систем завдяки використанню моделей штучного інтелекту, не зменшуючи при цьому ефективність виявлення точно позитивних кібератак на різних рівнях інфраструктури інформаційної системи. Для успішного досягнення мети даної роботи необхідно виконати наступні завдання:

1. Визначити основні компоненти інформаційних систем та дослідити їх особливості. Проаналізувати переваги та недоліки сучасних систем дослідження подій інформаційної безпеки. Проаналізувати властивості DevSecOps–підходу та його впливу на дослідження кіберзлочинів.

2. Дослідити проблеми та обмеження алгоритмів штучного інтелекту. Дане завдання включає порівняння властивостей алгоритмів ізоляційного лісу, випадкового лісу та алгоритмів машинного навчання між собою та аналіз можливостей GPT моделей для дослідження кіберзлочинів.

3. Розробити метод збору журналів подій з приманок на основі Blockchain, що можуть бути застосовані в інфраструктурі інформаційної системи, та дослідити кібератаки, спрямовані на приманки, за допомогою розробленої системи з метою перевірки її ефективності для виявлення кібератак та дослідження кіберзлочинів.

4. Розробити методологію дослідження кіберзлочинів для підвищення ефективності процесу виявлення кібератак та детального дослідження кіберзлочинів аналітиками інформаційної безпеки.

5. Розробити модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем, використовуючи алгоритми ізоляційного лісу, GPT та підхід DevSecOps.

6. Оцінити можливість використання системи дослідження кіберзлочинів для покращення систем менеджменту інформаційною безпекою, побудованих на основі міжнародного стандарту ISO/IEC 27001:2022 та відповідність системи вимогам ISO/IEC 27001:2022.

**Об’єктом дослідження** є система дослідження кіберзлочинів з використанням штучного інтелекту та підходу DevSecOps, що дозволяє визначати вразливості інфраструктури, виявляти кібератаки, спрямовані на інформаційну систему та аналізувати кіберзлочини на різних рівнях інформаційних систем.

**Предметом дослідження** є методи та засоби виявлення та дослідження кіберзлочинів в інформаційних системах та їх компонентах.

**Методи дослідження.** У процесі наукової роботи були застосовані такі методики: оптимізаційні процедури, імітаційне та аналітичне моделювання, методи математичної статистики, розробка на основі об'єктно-орієнтованого підходу, а також принципи теорії інформації та кодування.

**Наукова новизна** роботи полягає в тому, що:

1. Вдосконалено математичний апарат оцінки вразливостей інфраструктури інформаційних систем завдяки додаванню та обчисленню атрибутів досліджуваної інформаційної системи, а також впровадженню вагових коефіцієнтів. Це підвищило точність оцінки вразливостей, дозволяючи командам безпеки пріоритезувати виправлення вразливостей згідно з особливостями інформаційної системи.

2. Вперше розроблено метод збору журналів подій з приманок на основі технології Blockchain, що забезпечує децентралізацію даних. Розроблений метод

дозволив зменшити ризики спотворення та втрати даних під час зберігання журналів подій.

3. Отримав подальший розвиток математичний апарат виявлення кібератак завдяки впровадженню моделі ізоляційного лісу, GPT та DevSecOps–підходу Завдяки інтеграції можливостей виявлення аномалій ізоляційного лісу, властивостей обробки передбачуваної моделей GPT і цілісного фокусу безпеки DevSecOps, структура математичного апарату підвищила точність і швидкість виявлення кібератак.

4. Вперше розроблено модель комплексної системи дослідження кіберзлочинів, здатну виявляти та аналізувати кіберзлочини на різних рівнях інформаційної системи. Ця модель інтегрує моделі штучного інтелекту ізоляційний ліс, GPT та підхід DevSecOps, відрізняючись від традиційних систем дослідження подій інформаційної безпеки завдяки використанню комплексного підходу та інтеграції сучасних моделей та підходів інформаційної безпеки в єдину систему. Зокрема, використання ізоляційного лісу та GPT, а також систем аналізу вразливостей на різних рівнях розробки підвищує ефективність виявлення первинних причин кіберзлочинів та зменшує час реакції на атаки.

5. Вперше розроблено методологію дослідження кіберзлочинів, що використовує моделі ізоляційного лісу, GPT та DevSecOps–підхід. Дана методологія, на відміну від існуючих, виявляє кібератаки на різні рівні інфраструктури інформаційної системи, включно з атаками сканування, ін’єкціями шкідливого коду, атаками типу Directory Traversal та виявленням аномалій з порушенням логіки додатків, які можуть залишатися непоміченими класичними SIEM системами за відсутності поведінкових сигнатур, гарантуючи високий рівень безпеки даних.

**Практичне значення** одержаних результатів полягає у можливості їх безпосереднього застосування для покращення процесу управління інцидентами інформаційної безпеки у державних та приватних системах менеджменту інформаційної безпеки.

1. Розроблена методологія дослідження кіберзлочинів, що побудована на основі моделей ізоляційного лісу та GPT забезпечила відповідність процесу моніторингу інформаційної безпеки у системі менеджменту інформаційної безпеки контролю 8.16 міжнародного стандарту ISO 27001:2022. Впровадження методології дослідження кіберзлочинів у систему менеджменту інформаційної безпеки дало можливість виявляти кіберзлочини на ранніх їх стадіях, мінімізуючи ресурси, необхідні для забезпечення відповідності контролям 5.25 та 5.26 міжнародного стандарту ISO 27001:2022.

2. Впровадження системи дослідження загроз інформаційної безпеки як одного із елементів моделі дослідження кіберзлочинів забезпечило виявлення відомих кіберзагроз, користуючись публічними ідентифікаторами компрометації інформаційних систем. Використання DevSecOps–підходу та сканування інформаційних систем на різних рівнях інфраструктури вразливостей дало можливість корелювати вплив вразливостей інформаційної системи на кіберзлочини. Цей підхід дозволяє ідентифікувати відомі кіберзагрози за допомогою загальнодоступних ідентифікаторів, оптимізує процес усунення вразливостей шляхом сканування інфраструктури інформаційних систем та надає комплексне уявлення про стан безпеки інформаційної системи.

3. Експериментально підтверджено, що модель GPT–4.0 не лише точно визначає тип кіберзлочину, але забезпечує загалом щонайменше до 5% швидше виявлення кібератак, ніж GPT–3.5, що може мати вирішальне значення в реальних сценаріях, де час відповіді потрібно мінімізувати.

4. Розроблена модель з використанням ізоляційного лісу дала можливість зменшити час виявлення кібератак у середньому до 31% в повірянні з класичною SIEM–системою та виявляти невідомі атаки, що зумовлено здатністю навчання моделі ізоляційний ліс відрізняти нормальну поведінку від аномальної та працювати з аномаліями різного типу.

5. Експериментально визначено, що модель на основі GPT обробляє дані швидше, ніж це можливо для людини, ідентифікуючи закономірності та

взаємозв'язки, зменшуючи час дослідження кіберзлочинів у середньому до 60%, а для аномалій, що порушують логіку роботи додатку, до 7 разів.

Наукові та практичні результати виконаних досліджень використані у навчальному процесі кафедри захисту інформації Національного університету “Львівська політехніка”, зокрема для студентів спеціальності “125 – Кібербезпека” в курсі лекцій з дисципліни “Безпека програмного забезпечення”.

Основні результати дисертаційної роботи використано і впроваджено з метою покращення процесів виявлення інцидентів інформаційної безпеки компанією ТОВ “Бінарікс Україна”, реагування на інциденти інформаційної безпеки, компанією AG “Hiveon” та як елемент забезпечення відповідності заходам захисту стандарту ISO/IEC 27001:2022 для компанії ТОВ “Техмеджик”, що підтверджено актами впровадження.

**Особистий внесок.** Важливі наукові результати цієї дисертації були досягнуті автором незалежно. У роботах, опублікованих разом із співавторами, ключовий внесок належить автору. Зокрема, він зробив такий внесок (за нумерацією, вказаною у Додатку Б): [1, 5, 13, 16] - розробка методу дослідження кіберзлочинів з використанням Blockchain–технології; [3,4,7,8,11,14,17] - аналіз і дослідження існуючих методів протидії загрозам інформаційної безпеки на різних рівнях інфраструктури інформаційних систем; [6,12] – покращення методології дослідження кіберзлочинів шляхом інтеграції елементів DevSecOps підходу для протидії сучасним загрозам; [5,15] - поліпшення математичного апарату, який лежить в основі процесу дослідження кіберзлочинів; [9,10] - моделювання сучасних заходів захисту інформаційної безпеки для забезпечення відповідності міжнародним стандартам; [2,18] - дослідження використання моделей ШІ для дослідження кіберзлочинів.

**Апробація результатів.** Ключові наукові досягнення цієї дисертації були представлені та обговорювались на восьми міжнародних і вітчизняних науково-технічних конференціях та семінарах. Це участь у VII, VIII та IX Міжнародних науково-технічних конференціях "Захист інформації та безпека інформаційних систем" у Львові в 2019, 2021 та 2023 роках; участь у 15-ій Міжнародній

конференції IEEE з комп'ютерних наук та інформаційних технологій (CSIT) у Збаражі в 2020 році; участь у IV Всеукраїнській науково-практичній конференції молодих учених, студентів і курсантів у Львові в 2020 році; участь у IV Міжнародній конференції «Нові досягнення в галузі інформаційно-комунікаційних технологій» в 2021 році у Львові; участь у семінарі з забезпечення кібербезпеки в інформаційно-телекомунікаційних системах (CPITS) в 2021 у Києві, а також виступи на семінарах при вченій раді НАН України на тему: "Технічні засоби захисту інформації" у Львові в 2021 році. Окрім цього, дисертаційна робота була представлена на наукових семінарах кафедри захисту інформації Національного університету "Львівська політехніка".

**Публікації.** Основні результати дослідження викладено у вісімнадцяти наукових публікаціях, а саме: у десяти статтях (із них дев’ять – у фахових наукових виданнях України та одній – у періодичному виданні закордоном) і восьми тезах виступів на науково-практичних заходах.

***Структура та обсяг дисертації.*** Дисертація складається зі вступу, чотирьох розділів, що охоплюють 23 підрозділів, висновків, списку використаних джерел і додатків. Загальний обсяг дисертації становить 196 сторінок, з яких 157 – основний текст, 10– список використаних джерел (101 найменування), 29 – додатки.

# РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОБЛЕМИ РОЗРОБЛЕННЯ КОМПЛЕКСНОЇ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗАГРОЗ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ

## 1.1 Аналіз сучасного стану досліджень та публікацій

Щороку кількість кіберзлочинів у світі лише збільшується, а роль кібервійни стає доволі значною. Кіберзлочинність стає все більш поширеною, але відсутність консенсусу щодо того, що є кіберзлочинністю, має значний вплив на суспільство, правову й політичну реакцію та наукові дослідження [5]. У своїй роботі “Концептуалізація кіберзлочинності: визначення, типології та таксономії” Кірсті Філліпс та ін. підкреслюють критичну перешкоду: відсутність узгодженої термінології та рамок для ідентифікації та категоризації явища кіберзлочинності, що розвивається. Ця неоднозначність перешкоджає ефективним дослідженням, розробці політики та, щонайважливіше, зусиллям правоохоронних органів. Автори визнають обмеженість одиничних визначень, визначають існуючі спроби категоризації та класифікації кіберзлочинності за допомогою таксономії. Визнання цього ландшафту кіберзлочинів підкреслює необхідність більш комплексної концептуалізації кіберзлочинності.

Книга Крістофера Вайта та Брайана Мазанека “Розуміння кібервійни” описує еволюцію кіберконфлікту: від його коріння таємного шпигунства до складних цифрових наступів сьогодення. Вона аналізує, як держави та недержавні суб’єкти використовують кіберпотенціал для досягнення своїх цілей. “Розуміння кібервійни” визнає, що кіберконфлікт — це не одностороння справа. Книга визначає задіяних акторів: від національних держав, що володіють потужним цифровим арсеналом, до тіньових недержавних груп і навіть приватних компаній із власними стратегічними цілями [6]. Розуміння їхніх мотивів і можливостей має важливе значення для розбору кіберконфліктів і передбачення майбутніх загроз.

У зв’язку з актуальністю теми кіберзагроз відбулося величезне збільшення досліджень у сфері кібербезпеки для підтримки кіберпрограм та уникнення основних загроз безпеці, з якими стикаються ці програми [7]. У роботі “Загрози та вразливості кібербезпеки: систематичне дослідження” автори провели

систематичний аналіз дослідницьких статей, опублікованих у період між 2014 – 2019 роками. Вони використовували стратегію пошуку, щоб визначити відповідні статті, а потім проаналізувати їх на основі попередньо визначеного набору критеріїв. Це дозволило їм визначити найпоширеніші критичні загрози та вразливості кібербезпеки, а також зв'язки між ними. Дослідження виявило широкий спектр загроз кібербезпеці та вразливостей. Зокрема, це загрози, спрямовані на мережеву інфраструктуру, наприклад атаки на відмову в обслуговуванні, зловмисне програмне забезпечення та ботнети; загрози, спрямовані на програмні додатки, наприклад атаки з ін'єкцією SQL, атаки типу XSS та переповнення буфера; загрози, націлені на дані, наприклад порушення даних, несанкціонований доступ і втрата даних; загрози, спрямовані на операційну систему та апаратне забезпечення, наприклад атаки нульового дня, уразливості вбудованого програмного забезпечення та фізичні порушення безпеки. Результати цього дослідження мають важливе значення для фахівців з кібербезпеки. Розуміючи найпоширеніші критичні загрози та вразливості, організації можуть розробити ефективніші стратегії захисту.

Важливим аспектом протидії кібератакам після розуміння їх типів є правильно визначена стратегія захисту. У статті Х. Кеттані та П. Вейнрайт "Про найпоширеніші загрози кіберсистем" запропоновано аналіз, що включає загальні тенденції щодо складності атак, учасників, а також зрілість навичок і можливостей організацій для захисту від атак [8]. Автори підкреслюють концепцію дедалі складніших атак, організованих державними суб'єктами, в той час коли традиційні засоби захисту не в змозі їх виявляти. Х. Кеттані та П. Вейнрайт наголошують на необхідності автоматизації та проактивного аналізу загроз, щоб випереджати цих досвідчених супротивників, закликаючи організації застосовувати найсучасніші заходи безпеки та інвестувати в кваліфікований персонал.

Постійно зростаючий рівень складних високошвидкісних кібератак ставить перед людьми, які працюють у сфері кіберзахисту, нові виклики. У сфері розробки програмного забезпечення, що постійно розвивається, безпека більше не може бути просто прикрасою готового продукту [9]. Анна Коскінен у статті "DevSecOps:

створення безпеки в основі DevOps”  досліджує основні принципи DevOps – культуру, автоматизацію, вимірювання та обмін (CAMS). Авторка інтерпретує та застосовує ці принципи через контролі безпеки. Також подано тематичні дослідження, засвідчуючи успішне впровадження DevSecOps у різних організаціях. Ці приклади демонструють DevSecOps як підхід, що вбудовує статичний аналіз безпеки додатків (SAST), динамічний аналіз безпеки додатків (DAST) та аналіз складу програмного забезпечення (SCA) в основу розробки, що може призвести до створення не тільки більш безпечного, але й більш ефективного та стійкого програмного забезпечення. DevSecOps можна визначити як культурний підхід до покращення та прискорення досягнення цінності для бізнесу шляхом ефективної співпраці команд розробки, безпеки та операційної діяльності [10]. Стаття “Самообслуговування моніторингу кібербезпеки як активатор для DevSecOps” присвячена самообслуговуванню моніторингу кібербезпеки як інструменту впровадження практик безпеки в середовищі DevOps. Дане дослідження доводить, що інфраструктура моніторингу кібербезпеки із залученням підходу DevSecOps дозволила виявляти загрози, такі як атаки відмови, і допомогла краще передбачити проблеми спуфінгу. Цю інфраструктуру реалізовано відповідно до передових практик DevOps: вона автоматизована за допомогою сценаріїв і конфігураційних файлів, а її розгортання автоматизовано за допомогою технології віртуалізації та контейнеризації [11].

Спроба захистити дані, впроваджуючи передові практики, типу DevSecOps та технології захисту інформації, призвела до зростання ролі кібербезпеки. У зв’язку з постійно зростаючим ландшафтом загроз, наявним ризикам та проблемам, останні роки дослідження науковців фокусуються на ролі штучного інтелекту та його впливу на кібербезпеку у великих масштабах. Стаття “Машинне навчання для інтелектуального аналізу даних і автоматизації в кібербезпеці: поточні та майбутні перспективи” описує, як машинне навчання можна використовувати для автоматизації завдань кібербезпеки та покращення аналізу даних для виявлення загроз [13]. Дане дослідження дає можливість визначити, що машинне навчання революціонізує кібербезпеку, автоматизуючи завдання та надаючи глибше

розуміння даних безпеки. Сахніні Джейкоб у праці “ШІ та безпека критичної інфраструктури” досліджує використання штучного інтелекту для захисту критичної інфраструктури, як-от електромережі та транспортні системи, від кібератак. Дане дослідження підтверджує, що штучний інтелект може бути цінним інструментом для захисту критичної інфраструктури, але він також створює нові вразливості, які потрібно усунути [15]. Актуальність використання машинного навчання також підкреслено у статті “Штучний інтелект і міжнародна безпека” Майкла Горовіца. Проте автори аналізують вплив штучного інтелекту (ШІ) на міжнародну безпеку, включаючи потенційні переваги та ризики, а також вказують, що штучний інтелект має потенціал для покращення як наступальних, так і оборонних можливостей у війні, що викликає занепокоєння щодо нової гонки озброєнь [14].

Дані дослідження вказують, що штучний інтелект може мати як позитивний, так і негативний вплив на безпеку організації, зокрема Шуай Чжоу у статті “Змагальні атаки та захист у глибокому навчанні: з точки зору кібербезпеки” досліджує, як зловмисники можуть маніпулювати моделями глибокого навчання (наприклад, розпізнаванням обличчя) на свою користь і як захиститися від цих атак. У ході даного дослідження автор дійшов до висновку, що змагальні атаки становлять серйозну загрозу безпеці глибокого навчання, але розробляють різні механізми захисту.

Не зважаючи на те, що штучний інтелект може покращити стан кібербезпеки, його природа вимагає обережності. Як підкреслено у проаналізованих матеріалах, інноваційні підходи можуть підвищити автоматизацію завдань, аналізу даних і навіть допомогти запобіганню кібератакам. Однак потреба у комплексному підході до дослідження кіберзлочинів може допомогти з проблемами у сфері міжнародної безпеки та критичної інфраструктури, зважаючи на потенційну гонку озброєнь та використання зловмисниками. Зрештою, як підхід DevSecOps, так і моделі штучного інтелекту, виступають у ролі трансформаційної сили в кібербезпеці, що вимагає відповідального розвитку, розгортання та подальшого дослідження, щоб отримати переваги та зменшити ризики [12].

## 1.2 Огляд сучасних компонентів інфраструктури інформаційних систем та пов’язаних з ними загроз безпеки

На сьогодні не існує єдиного загальноприйнятого визначення терміну “інформаційна система”. Можна стверджувати, що визначення цього поняття є складним завданням і його точний опис можливий лише з використанням різних підходів та поглядів. Одне із офіційних визначень, яке надає закон України 80/94-ВР “Про захист інформації в інформаційно-телекомунікаційних системах”, описує інформаційну систему як організаційно-технічну систему, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів [16].

Також в українському законодавстві визначено й інші альтернативні тлумачення терміну, зокрема:

1. Інформаційна система – автоматизована система, комп'ютерна мережа або система зв'язку [16].
2. Інформаційна система – організаційно-технічна система обробки інформації за допомогою технічних і програмних засобів [17].
3. Інформаційна система — система, призначена для одержання, обробки, зберігання, відображення та/або реєстрації даних про технічний стан конструкцій, систем, елементів, їх властивості та/або функціонування [18].

Усі ці визначення вказують на автоматизацію обробки даних, що є важливим аспектом роботи системи. Є три основні способи, за допомогою яких інформаційні системи можуть значно вплинути на успіх організації [19]. Зокрема, це покращення комунікації всередині організації, встановлення зв’язку між організацією та її зацікавленими сторонами, а також допомога людям в організації приймати кращі рішення.

Важливо розуміти загальну концепцію інформаційних систем. Для побудови глобальних розподілених інформаційних систем використовують архітектуру інтеграції інформаційно-обчислювальних компонентів на основі об'єктно-орієнтованого підходу [20].

Враховуючи принцип декомпозиції, зазвичай практикується розробка

інформаційних систем з функціональною декомпозицією їх компонентів, що означає створення багаторівневої структури. Можна виділити три основні функціональні групи для вирішення різних завдань: сервіси, що відповідають за взаємодію з користувачами, сервіси бізнес-логіки та група сервісів, що керує ресурсами системи.

Для реалізації такого функціонального розподілу необхідно створювати програмну систему з багаторівневою архітектурою. Зокрема, компонент, відповідальний за інтерфейс з користувачем, обробляє дії користувача, такі як натискання клавіш, навігацію між різними контролями і відображення інформації, забезпечуючи зручний інтерфейс для користувача. Прикладний компонент є набором алгоритмів реалізації функцій системи. Компонент управління ресурсами відповідає за обробку, зберігання та передачу даних. Для спрощеної схеми роботи інформаційної системи представлено схему роботи інформаційної системи (рис. 1.1).

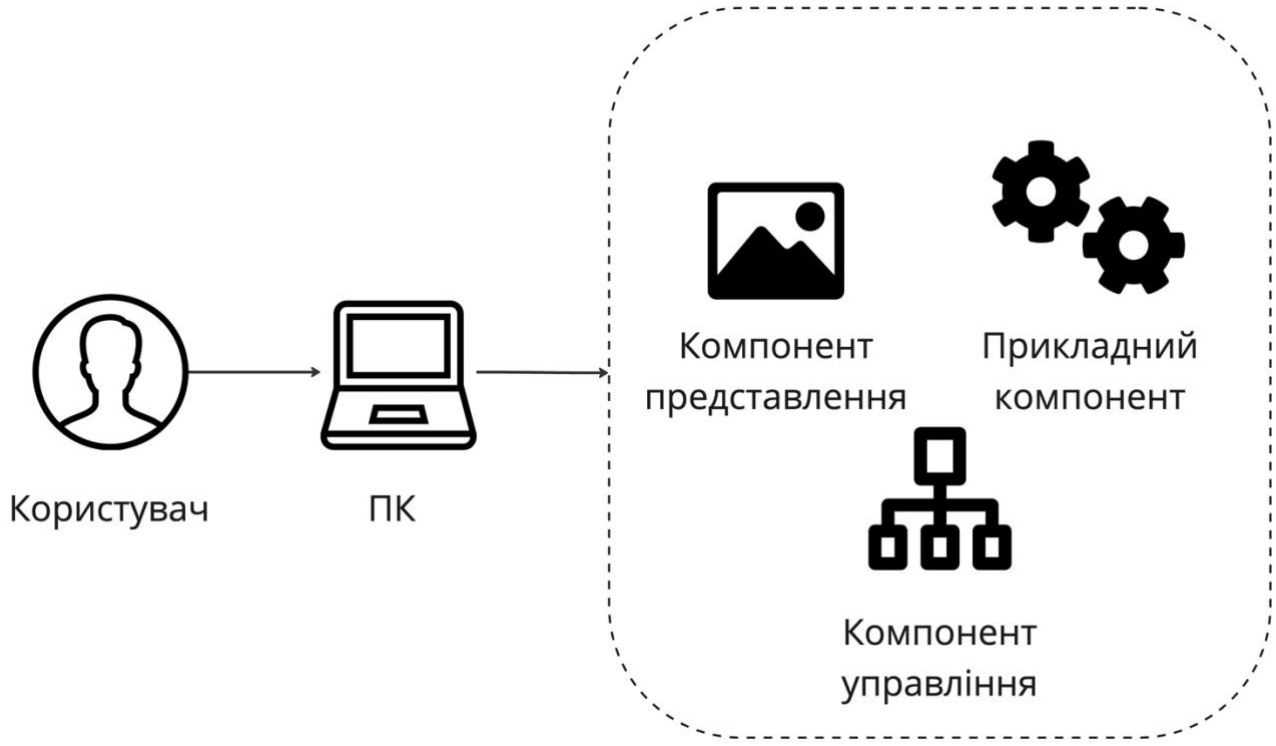



Рис. 1.1. Компоненти інформаційної системи

Для роботи з інформаційними системами та їх інфраструктурою, потрібно розуміти основні елементи, які складають сучасну інформаційну систему, і їхнє значення та кваліфікацію. Кваліфікація компонентів відноситься до ідентифікації

та оцінки компонентів, потенційно здатних відповідати системним вимогам [21].

Загальноприйнятим є поділ інформаційної системи на наступні групи: апаратні та програмні компоненти, мережеве обладнання, хмарні сервіси, елементи керування даними та сервіси безпеки. Детальний аналіз кожної з груп та їх компонентів для цього дослідження дасть можливість сформувати експериментальні атаки на різні компоненти інфраструктури інформаційних систем.

До апаратних компонентів зараховують сервери, пристрої зберігання даних та комутаційне обладнання. Сервери можуть бути як фізичними пристроями, так і віртуалізованими рішеннями. Вони виконують програми, розміщують бази даних і надають послуги, забезпечуючи доступність критичних ресурсів. Організації використовують різноманітні рішення для зберігання даних, від традиційних жорстких дисків до високошвидкісних накопичувачів (SSD) і мережевих пристроїв зберігання даних (NAS). Ці пристрої надійно зберігають дані, забезпечуючи швидкий доступ і пошук. До мережевого обладнання зараховують: маршрутизатори, комутатори та брандмауери.

Наступною групою є програмні компоненти. Операційні системи є основою інфраструктури, керуючи апаратними ресурсами та надаючи основні послуги. Системи керування базами даних (СУБД), для прикладу MySQL і Oracle, забезпечують структуроване зберігання та пошук даних. Програмне забезпечення проміжного рівня з’єднує різні програми, дозволяючи їм ефективно спілкуватися. Технології віртуалізації та контейнеризації, такі як VMware і Docker, оптимізують використання ресурсів та ізоляцію програм.

Мережева інфраструктура включає локальні мережі (LAN), які з’єднують пристрої в певних місцях, глобальні мережі (WAN), що з’єднують кілька локальних мереж у різних географічних областях, та хмарні мережеві рішення, які пропонують такі постачальники, як AWS і Azure, для розширення мереж у хмару.

Хмарні послуги надаються у вигляді трьох основних моделей сервісів. Інфраструктура як послуга (IaaS) пропонує віртуалізовані обчислювальні ресурси, платформа як послуга (PaaS) забезпечує платформу розробки та розгортання, а

програмне забезпечення як послуга (SaaS) надає програмні додатки через Інтернет, зменшуючи вимоги до локального обслуговування.

Зберігання та керування даними охоплюють бази даних та сховища даних. Бази даних, як реляційні, так і NoSQL, зберігають структуровані та неструктуровані дані та керують ними.

Та основною групою, яку варто виділити, є компоненти безпеки, що включають брандмауери для захисту мережі, системи керування ідентифікацією та доступом (IAM) для контролю доступу користувачів, шифрування для захисту даних під час передачі та у стані спокою, а також керування інформацією та подіями безпеки (SIEM) для подій безпеки в реальному часі.

Дане дослідження також пропонує розглянути вихідний код як окрему групу. Вихідний код системи насправді є важливим компонентом інфраструктури сучасної інформаційної системи, але його зазвичай не класифікують, як саму інфраструктуру. Натомість вихідний код вважають частиною програмного рівня в інфраструктурі, оскільки він представляє інструкції, які керують поведінкою програми. Це критично важливий компонент будь-якої інформаційної системи, адже він визначає функціональність, логіку та поведінку програмного забезпечення.

Підсумовуючи, сучасна інфраструктура інформаційної системи є складною та багатогранною екосистемою, тому розуміння основних компонентів інфраструктури є ключовим при аналізі подій безпеки з різних пристроїв та сервісів.

### 1.1.1 Порівняльна характеристика інфраструктури інформаційної системи, побудованої на основі апаратних рішень та хмарних технологій

Для побудови інформаційної безпеки сучасний бізнес може вибирати між локальною серверною інфраструктурою та хмарними рішеннями, включаючи програмне забезпечення, сервери, сховища, резервні копії та безпеку. Хмара стає все більш популярною, але локальні реалізації зберігають важливу роль. Кожне

рішення має свої переваги та недоліки.

Модель “on-premise” – це традиційна модель обчислень для підприємств. Програмне забезпечення з використанням моделі “on-premise” встановлюють у внутрішній системі організації разом із апаратним забезпеченням та іншою інфраструктурою, необхідною для його функціонування [22]. У цій реалізації все апаратне та програмне забезпечення знаходиться в приміщеннях організації. Бізнес купує та утримує власні сервери, розташовані у безпечному приміщенні з контролем за кліматом. У даному випадку компанія потребує IT-підтримки для керування власним обладнанням, а також для обслуговування відповідних систем HVAC (вентиляція, опалення, кондиціювання повітря), щоб підтримувати обладнання в робочому стані. Також організація повинна забезпечити процес оновлення програмного забезпечення та регулярно виконувати резервне копіювання. З ростом бізнесу необхідно оновлювати обладнання відповідно до щораз більших потреб.

За останні роки хмарні обчислення стали хорошою альтернативою локальній інфраструктурі. Оплачуючи підписку за доступ до центрів обробки даних, компанія може зберігати свої дані на невеликій частині віддалених серверів. Хмарний провайдер піклується про обслуговування, резервне копіювання, оновлення програмного забезпечення, живлення та HVAC. У цьому випадку клієнт покладається лише на власні робочі станції та з’єднання Інтернет.

Хмарні обчислення – це модель забезпечення повсюдного та зручного мережевого доступу до загального пулу конфігурованих обчислювальних ресурсів (наприклад, мереж передачі даних, серверів, пристроїв зберігання даних, додатків і сервісів – як разом, так і окремо), які можуть бути надані і розгорнуті з мінімальними експлуатаційними витратами і/або зверненнями до провайдера [22]. Згідно з NIST спеціальної публікації 800-145, хмарні обчислення – це модель для забезпечення повсюдного, зручного мережевого доступу на вимогу до спільного пулу сконфігурованих обчислювальних ресурсів (наприклад, мереж, серверів, сховищ, програм та послуг), які можна швидко забезпечити та випустити з мінімальними зусиллями управління або взаємодією з постачальником послуг [23].

Визначення ISO/IEC дуже подібне. Хмарні обчислення – це парадигма надання мережевого доступу до масштабованого та еластичного пулу спільних фізичних або віртуальних ресурсів із забезпеченням самообслуговування та адмініструванням на вимогу [24].

Основними техніками створення хмари є абстракція та оркестрація. Ресурси абстрагуються від локальної серверної інфраструктури і забезпечується автоматизація для координації та доставлення набору ресурсів споживачам. У цьому полягає різниця між хмарними обчисленнями та традиційною віртуалізацією; віртуалізація абстрагує ресурси, але, як правило, їй не вистачає оркестрації, щоб об’єднати їх і доставити клієнтам на вимогу, а не покладатися на ручні процеси [25].

Є 3 основні моделі надання хмарних послуг: SaaS, PaaS та IaaS. Для візуалізації їх відмінності наведено рис. 1.2. Додатки SaaS (Програмне забезпечення як послуга) – це складні багатокористувацькі платформи, які часто створюються на основі IaaS і PaaS для підвищення продуктивності та стабільності [25]. Ці програми часто пропонують загальнодоступні API для підтримки різних клієнтів, таких як веб-браузери та мобільні програми. PaaS (платформа як послуга) орієнтована на розробників і пропонує хмарне середовище для створення програм [25]. На відміну від SaaS, який призначений більше для кінцевого користувача, цей варіант призначений для розробників. PaaS охоплює широкий спектр послуг, включно з базами даних і обмінами повідомленнями, що спрощує розробку та обслуговування додатків. Користувачі взаємодіють із платформою та захищені від базової інфраструктури. Інфраструктура як послуга (IaaS) – це модель хмарних обчислень, яка спирається на фізичні засоби та обладнання [25]. Ці ресурси абстрагуються та об’єднуються за допомогою таких методів, як віртуалізація та оркестровка. API відіграють центральну роль в управлінні цими ресурсами, і багато хто використовує для зв’язку REST через HTTP. Інтерфейси керування хмарою дозволяють користувачам віддалено налаштовувати та контролювати свої хмарні ресурси. Захист рівня керування хмарою є головним пріоритетом, оскільки неавторизований доступ може призвести до компрометації всього розгортання

хмари. По суті, IaaS містить фізичну інфраструктуру, абстракцію, автоматизацію та API для управління ресурсами.

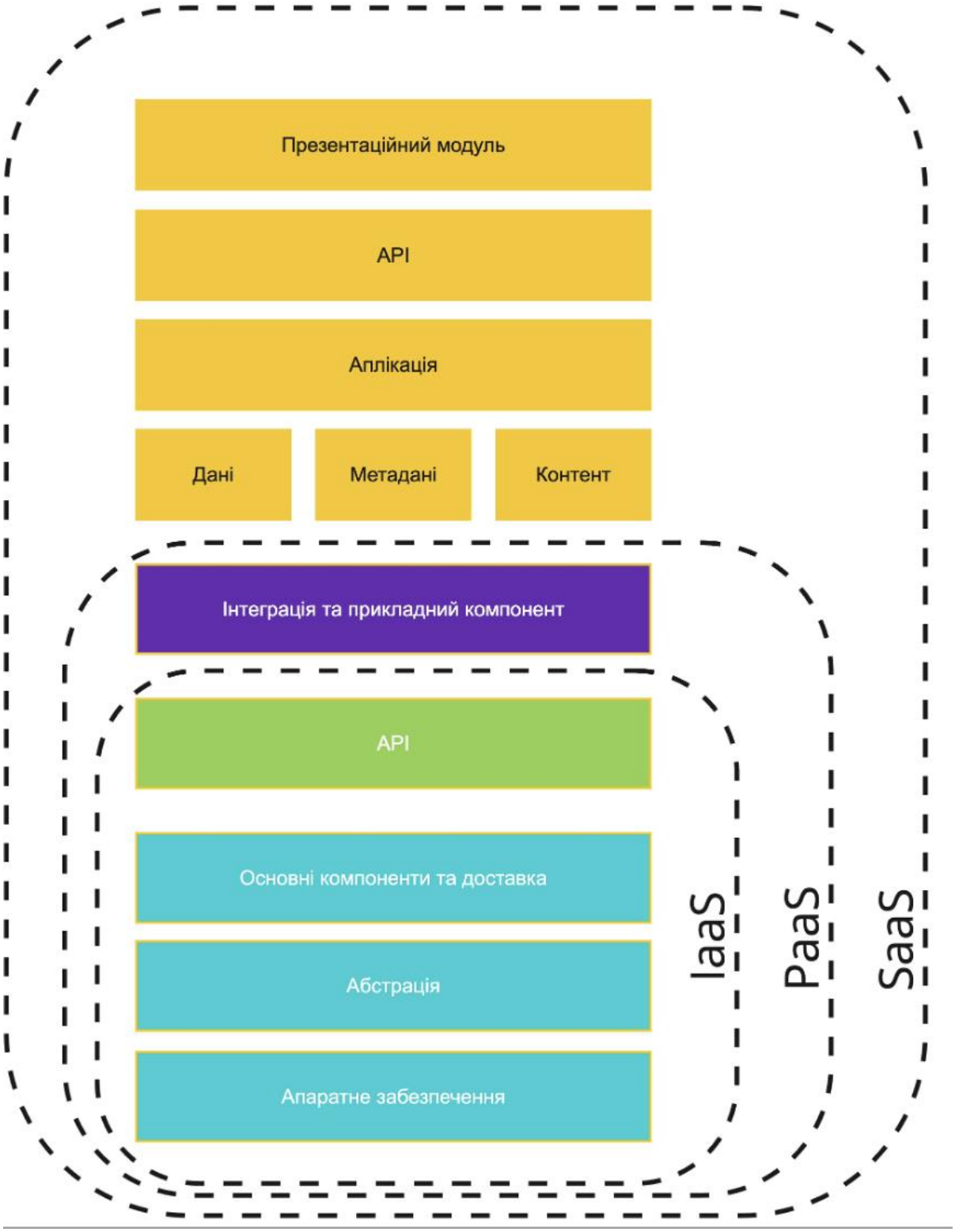



Рис. 1.2. Основні моделі надання хмарних послуг

Проаналізувавши моделі надання хмарних послуг та особливості інфраструктури “on–premise”, було сформовано порівняння характеристик інфраструктури інформаційних систем у таблиці 1.1

Таблиця 1.1.

Порівняння інфраструктури інформаційної системи, побудованої на основі апаратних рішень та хмарних технологій

| **Характеристика** | **Інфраструктура інформаційної системи типу On–Premise** | **Хмарна інфраструктура** |
|---|---|---|
| Місцезнаходження | Фізично в приміщеннях організації (наприклад, центри обробки даних). | Розміщується за межами організації в сторонніх центрах обробки даних. |
| Структура витрат | Закупівля обладнання. Постійні інвестиції в оновлення та заміну обладнання. | Плата за використання або модель витрат на основі підписки. |
| Масштабованість | Обмежена масштабованість; потребує придбання та встановлення обладнання. | Швидка і гнучка масштабованість; ресурси можна регулювати за потреби. |
| Технічне обслуговування та оновлення | Внутрішні IT-групи відповідають за обслуговування, оновлення та безпеку. | Хмарний постачальник займається обслуговуванням інфраструктури, оновленнями та виправленнями безпеки. |

Продовження таблиці 1.1.

| | | |
|---|---|---|
| Розподіл ресурсів | Статичний розподіл ресурсів, що призводить до потенційного невикористання. | Динамічний розподіл ресурсів, оптимізація використання та ефективність витрат. |
| Безпека та відповідність | Повний контроль і відповідальність за заходи безпеки та відповідності міжнародним стандартам. | Спільна відповідальність із хмарним постачальником за безпеку; провайдер забезпечує безпеку інфраструктури. |
| Резервування та аварійне відновлення | Зазвичай від організацій вимагається планування та впровадження резервування та аварійного відновлення. | Хмарні провайдери часто пропонують вбудовані резервування та варіанти аварійного відновлення |
| Глобальне охоплення | Обмежується фізичними місцями розташування організації. | Глобальна присутність, що забезпечує доступ у всьому світі. |

Отож, при виборі інфраструктури для побудови інформаційної системи організація повинна розуміти основні відмінності та переваги кожної з них. Проте

основним активом, що становить цінність для організації, є інформація, яку обробляє інформаційна система, тому необхідно провести детальний аналіз загроз для інформаційних систем, побудованих на хмарних рішеннях та локальній інфраструктурі.

### 1.1.2 Проведення аналізу відмінностей загроз інформаційної безпеки для інформаційних систем, побудованих на інфраструктурі типу “on-premise” та хмарній інфраструктурі

Загроза – це можлива небезпека, яка може використати вразливість для порушення безпеки і завдати можливої шкоди. Вона може бути навмисною, випадковою або спровокованою іншим чином обставиною, здатністю, дією чи подією [26].

Для розуміння мотивації потрібно розглянути одні з найвідоміших витоків інформації та кібератак останніх років. У 2019 році понад 540 мільйонів записів користувачів Facebook були зламані та опубліковані в службі хмарних обчислень Amazon. Основною причиною цього інциденту стали незахищені резервні копії, які були загальнодоступними на AWS без будь-якого механізму контролю доступу.

8 грудня 2020 р. провідна компанія з кібербезпеки FireEye оголосила, що була зламана групою державних хакерів. У рамках цієї атаки зловмисники навіть викрали внутрішні розробки експертів FireEye. Microsoft, FireEye, SolarWinds та уряд США опублікували скоординований звіт про те, що SolarWinds було зламано групою державних хакерів, а FireEye постраждала як один із клієнтів SolarWinds.

У січні 2021 року представники Міністерства юстиції США підтвердили, що вони також постраждали від злому SolarWinds. До того ж, агентство було однією з небагатьох жертв, яких хакери продовжували атакувати та, зрештою, отримали доступ до внутрішніх поштових скриньок.

Отже, аналізуючи дані події та загальні тенденції кібербезпеки, можна вказати, що мотивація сучасних атак – це фінансова вигода, отримана з продажу викрадених даних або ж інтелектуальної власності компаній, яку спричинила фінансова криза у світі. Хакери можуть скомпрометувати організаційну

інфраструктуру або отримати несанкціонований доступ до даних чи інформації, які можуть бути використані для отримання фінансової вигоди. Крім того, хакери порушують або саботують виробництво електроенергії тощо, створюючи хаос і анархію [27]. Коли мотивація визначена, необхідно зрозуміти, як виконується атака. Для цього можна використовувати ланцюжок кібератаки, показаний на Моделі Cyber Kill Chain [28] – (Рис. 1.3), для аналізу вектору атаки в будь-який момент атаки зловмисника.

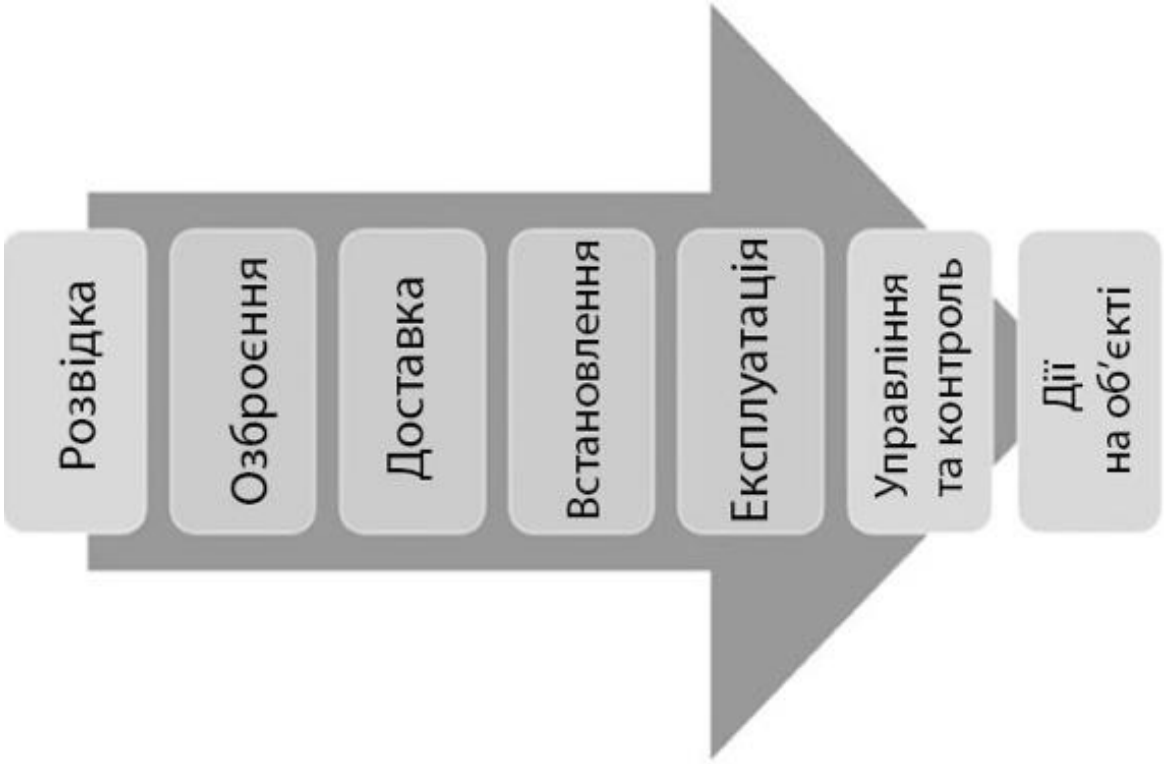



Рис. 1.3. Модель Cyber Kill Chain

До прикладу, найпоширенішими та найпопулярнішими нападами під час пандемії COVID-19 була група атак соціальної інженерії, яка використовувала страхи людей або співчуття [28]. Даний вектор атаки представлено схематично на Рис. 1.4 згідно з моделлю Cyber Kill Chain. Цю атаку зловмисники реалізували шляхом підроблених версій благодійних веб-сайтів, щоб отримати кошти від жертви. Зловмисники використовували шкідливе програмне забезпечення Alozurt. Хакери досліджували найпоширеніші страхи кінцевих користувачів – необхідність інформації про COVID-19. Найпопулярнішими ресурсами, які відвідували

користувачі під час пандемії, були онлайн-інтерактивні карти COVID-19, які ділилися інформацією про стан зараження в різних країнах. Другим джерелом інформації для людей були новинні ресурси. Тож зловмисники використали страхи людей і створили тисячі підроблених карт COVID-19 і фейкових новинних сайтів для поширення шкідливого програмного забезпечення, експлуатації ним вразливості CVE-2017-11882 та керування скомпрометованим активом жертви.


Рис. 1.4. Дослідження вектору атаки – соціальна інженерія (COVID-19)

Популярним вектором атак, яким користуються сучасні зловмисники, є атаки з використанням ланцюжка постачання. Безперервність бізнесу під час пандемії було забезпечено високоякісними можливостями дистанційної роботи, де це можливо, та за використання послуг сторонніх організацій. Протягом березня-

квітня 2020 року програмне забезпечення Zoom було основною метою для зловмисників. Щодня системи дослідження загроз інформували щодо проблем безпеки та конфіденційності програмного забезпечення Zoom. Близько 500 000 облікових записів Zoom було продано на хакерських форумах. У додатках Zoom виявлялись численні критичні вразливості, та як наслідок хмарний сервіс Zoom, який зберігав записані конференції, був скомпрометований [29]. У цей час найпоширенішою проблемою було Zoombombing, що дозволяло зловмисникам приєднуватися до незахищених зустрічей Zoom, а найгіршим сценарієм у цьому випадку було зараження шкідливим програмним забезпеченням користувачів через чати Zoom.

Також важливо розглянути атаку на ланцюг постачання з використанням сторонніх постачальників послуг. Даний вектор атаки зображено на Рис. 1.5. До прикладу, в квітні 2020 року було зламано 25 000 акаунтів Всесвітньої організації охорони здоров'я фонду Б. Гейтса. Як повідомили вказані організації, атака проводилася на сторонні сервіси, де оброблялись облікові записи користувачів [30].

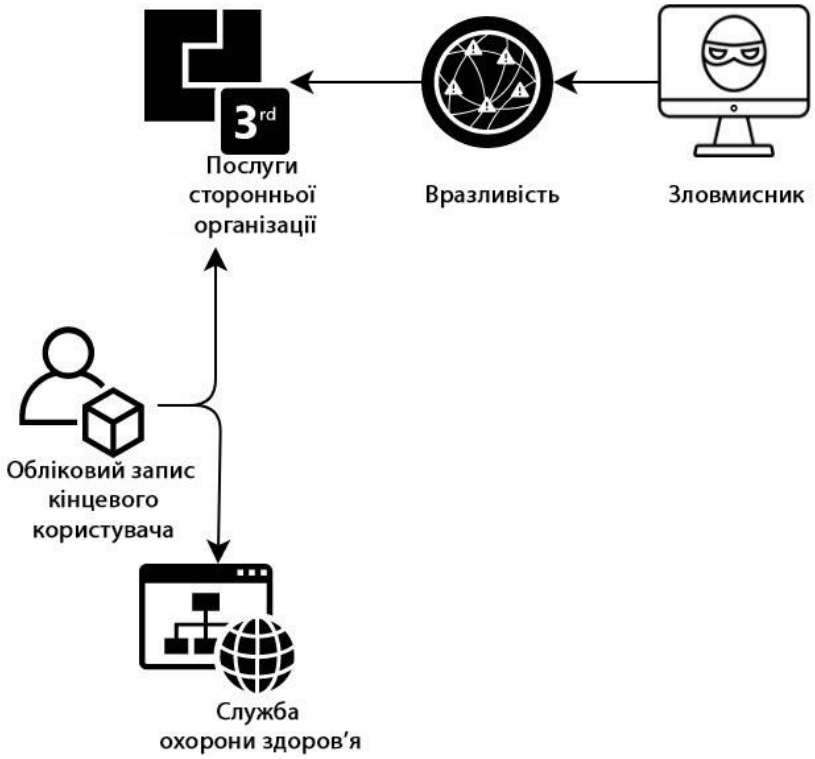



Рис. 1.5. Дослідження вектору атаки – постачання послуг (COVID-19)

Дослідження сучасних векторів атак, подій інформаційної безпеки та мотивації зловмисників дає можливість оцінити сучасні загрози безпеки. Організації стикаються з цілою низкою загроз безпеці, які здебільшого пов’язані з фізичними та операційними аспектами керування власними центрами обробки даних або серверними об’єктами. Цінність центру обробки даних формується в його ядрі. Самі дані та можливість їх обробки на вимогу є цілями, що представляють великий інтерес для зловмисників. Існує багато можливих шляхів, якими зловмисники можуть піти для досягнення цих цілей. Пропонований список формує основні загрози безпеці організацій, що використовують інфраструктуру on-premise:

1. Однією з важливих проблем є фізична безпека, коли несанкціонований доступ до цих об’єктів може призвести до витоку даних, фальсифікації обладнання або збоїв у роботі.

2. Шкідливе програмне забезпечення становить постійний ризик, особливо якщо локальні системи не захищені належним чином і не оновлюються.

3. Витік даних як із внутрішніх, так і зовнішніх джерел є значною загрозою, оскільки організації несуть виняткову відповідальність за впровадження надійних заходів безпеки.

4. Втрата даних також є загрозою, пов’язаною із збоями обладнання або невідповідними стратегіями резервного копіювання.

5. Компрометація облікового запису через слабкі паролі або неправильне керування обліковими даними користувача може призвести до неавторизованого доступу. Внутрішні та зовнішні загрози, навмисні чи ненавмисні, за браком належного навчання персоналу, можуть виникати від співробітників або осіб, які мають доступ до локальних систем.

6. Зрештою, підтримка належного керування виправлення вразливостей має вирішальне значення для усунення вразливостей, оскільки організації повинні регулярно оновлювати та захищати своє локальне обладнання та програмне забезпечення.

Ці загрози підкреслюють унікальні проблеми, з якими стикаються організації, які керують своєю інфраструктурою в традиційних локальних умовах.

Якщо сфокусуватись на інформаційних системах, побудованих у хмарі, то критичною загрозою безпеки хмарних рішень є відсутність чітко визначеної архітектури та стратегії безпеки хмари. Динамічні обчислення та служби зберігання хмарних обчислень сприяють значним змінам у сфері інформаційних технологій, водночас це також завдало величезного впливу на безпеку та конфіденційність, а саме:

1. Технологія віртуалізації та відповідна розподілена модель може спричинити втрату даних на одному фізичному пристрої.

2. Існує брак довіри між користувачем і хмарною платформою, тому користувачі не розуміють, як дані використовуються в хмарній платформі.

Саме тому система безпеки повинна узгоджуватися з бізнес-цілями та завданнями організації, правильно оцінювати відповідальність користувачів та для кожного типу моделі хмарного сервісу має бути встановлено регулярне моделювання загроз і постійний моніторинг. Ці заходи можуть допомогти організаціям створити безпечну архітектуру для своєї хмарної інфраструктури та вчасно виявити загрози. Проте для різних моделей надання хмарних послуг можна виділити відповідні загрози.

Оскільки використання SaaS зростає, експерти відкривають нові загрози щодня. Загрози SaaS можуть виникати як від постачальника, так і SaaS архітектури. Найбільш поширеними загрозами SaaS, створеними постачальником, є витік даних, зумовлений помилкою постачальника, приховування неправильної поведінки клієнтом, повторне використання ресурсів та ненадійні обчислення клієнта. До загроз, що походять від SaaS архітектури, відносять загрози доступу до мережі, маніпуляції спільними ресурсами, проблеми автентифікації, ненадійні постачальники та неповне видалення даних [31].

Загрози, що стосуються PaaS, пов’язані з особливістю моделі. Зокрема, це несанкціонований доступ до веб-порталу/консолі та інтерфейсу керування, атаки можливі з використанням веб-інтерфейсу, зокрема XSS атаки, SQL, підробка міжсайтового запиту, атаки грубої сили тощо, шкідливе програмне забезпечення та використання вразливих публічних сервісів [32].

Для IaaS інфраструктури можна віднести ті ж загрози, що й для локальної інфраструктури, окрім проблем фізичної безпеки. Проте серед поширених локальних загроз можна виокремити DDoS-атаки, видобуток цифрової валюти, шкідливе програмне забезпечення, атаки типу Bruteforce з метою викрадення облікових даних і використання вразливостей у застарілому програмному забезпеченні [33].

Проаналізувавши загрози інфраструктури інформаційних систем, побудованих на локальній та хмарній інфраструктурі, було сформовано їх порівняльну характеристику у таблиці 1.2

Таблиця 1.2.

Порівняння загроз локальної та хмарної інфраструктури інформаційної системи

| Міркування безпеки | Локальна інфраструктура | Хмарна інфраструктура |
|---|---|---|
| Фізична безпека | Організації повинні захищати власні центри обробки даних або серверні кімнати. | Хмарний провайдер керує фізичною безпекою в центрах обробки даних, знижуючи ризик несанкціонованого доступу. |
| Шкідливі програми та вірусні атаки | Вразлива, якщо не захищена належним чином і не оновлюється. | Хмарні служби часто включають вбудовані засоби безпеки для захисту від зловмисного програмного забезпечення та вірусів, але конфігурації клієнта можуть створювати ризики. |

Продовження таблиці 1.2.

| | | |
|---|---|---|
| Витік даних | Висока ймовірність витоку даних від внутрішніх або зовнішніх загроз; організація у даному випадку несе повну відповідальність за заходи безпеки. | Результат неправильної конфігурації, скомпрометованих облікових записів або неправильного керування доступом; спільна відповідальність між хмарним постачальником і клієнтом. |
| Злом облікового запису | Неавторизований доступ може статися через слабкі паролі або скомпрометовані облікові записи користувачів. | Компрометація хмарного облікового запису може бути наслідком слабких паролів, фішингу або неправильного керування доступом. |
| Втрата даних | Ризик втрати даних через апаратні збої або відсутність стратегії резервного копіювання. | Дані можуть бути втрачені в хмарі через випадкове видалення або неправильне керування хмарним сховищем. |
| Внутрішні загрози | Інсайдери з доступом можуть створювати загрози безпеці, навмисно чи ненавмисно. | Інсайдерські загрози, застосовні як до локальних, так і до хмарних середовищ, за участю співробітників або користувачів зі зловмисними намірами або помилками. |

Продовження таблиці 1.2.

| | | |
|---|---|---|
| Керування процесом виправлення вразливостей | Відповідальність організації за виправлення вразливостей та оновлення систем. | Хмарний постачальник керує виправленням вразливостей для базової інфраструктури, а клієнти керують конфігураціями та додатками. |
| Спільна відповідальність | Організація несе повну відповідальність за безпеку всіх компонентів інфраструктури. | Хмарна безпека – це спільна відповідальність між постачальником (для інфраструктури) і клієнтом (для даних, конфігурацій і програм). |

Отже, загрози безпеці в хмарній інфраструктурі відрізняються від загроз у локальній інфраструктурі через модель спільної відповідальності. У хмарі існує розподіл відповідальності: постачальник керує базовою безпекою інфраструктури, а клієнти відповідають за безпеку своїх даних, програм і конфігурацій. Така домовленість може призвести до розкриття даних через неправильно налаштовані сервіси, незахищені API та слабкі засоби контролю доступу, а також ризик компрометації облікового запису. Крім того, глобальне охоплення хмарних сервісів може розширити потенційну поверхню атаки. Водночас локальна інфраструктура вимагає зосередження на фізичній безпеці, оскільки організації повинні захищати свої центри обробки даних або серверні кімнати від несанкціонованого доступу, втручання або крадіжки. Також загрози локальної інфраструктури включають зловмисне програмне забезпечення та атаки шкідливим програмним забезпеченням, витоки даних, втрату даних та при цьому організації несуть повну відповідальність за дотримання вимог безпеки. Ці відмінності підкреслюють важливість індивідуальних стратегій безпеки на основі обраної моделі інфраструктури. До того ж, необхідно забезпечити процес постійного дослідження та аналізу події безпеки, використовуючи різні системи моніторингу журналів

подій, їх кореляції між собою та вести аналіз даних розвідки про загрозу. Також організації повинні забезпечувати постійне покращення як інфраструктури, так і основних компонентів інформаційних систем, зокрема впроваджуючи процеси безпечної розробки інформаційних систем.

### 1.3 Характеристика та аналіз використання процесу DevSecOps

Останнім часом технологічна галузь стала свідком частого компрометування даних, що загрожує функціям безпеки та конфіденційності даних користувачів через затримку виявлення вразливостей і загроз. Є декілька факторів, що дозволяють зловмисникам використовувати ці прогалини в безпеці.

Згідно зі звітом DBIR 2020, 43% витоків даних минулого року були співвіднесені з уразливими місцями, пов'язаними з інформаційними системами. Це корелюється насамперед з тим, що в багатьох підходах розробників відсутні надійні заходи безпеки, щоб перешкодити зловмисникам, які можуть обійти обмеження, що призводить до значного збільшення витоку даних.

Традиційний процес розробки програмного забезпечення, відомий як життєвий цикл розробки програмного забезпечення (SDLC), зазвичай починається з дослідження вимог до програми та цільового користувача або ринку. Основний акцент у SDLC роблять на швидкій розробці багатофункціональних додатків або програмного забезпечення, що керує даними з метою захопити ринок і досягти швидкого повернення інвестицій (ROI). Ця вимога охоплює різні аспекти, зокрема функціональність програми, дизайн і взаємодію з користувачем. Це передбачає комплексне планування, яке охоплює такі аспекти: фінансовий бюджет програми, зовнішній вигляд, макет, архітектурні рішення, зберігання та передача даних, взаємодія з користувачем та інтеграція з іншими системами чи мережами. Щоб ефективно запобігти порушенням і зміцнити захист, організації повинні вжити заходів безпеки в процес розробки інформаційних систем. Є багато вагомих причин інтегрувати безпеку в життєвий цикл розробки програмного забезпечення (SDLC), як-от: запобігання витоку даних, мінімізація наслідків зламу чи кібератаки, виявлення вразливостей на етапі розробки інформаційної системи, захист репутації та збереження довіри зацікавлених сторін. Це може бути забезпеченим підходом

DevSecOps. Підхід DevSecOps є спробою створити та включити сучасні методи безпеки, які можна впровадити у швидкий та гнучкий світ DevOps [34]. Він сприяє розширенню мети DevOps щодо стримулювання співпраці між розробниками та операторами, також із залученням експертів із безпеки з початку [35]. DevSecOps– методологія також враховує включення вимог інформаційної безпеки під час розробки, проведення аналізу загроз для програми, здійснення перевірок безпеки протягом усього процесу розробки та розслідування інцидентів, пов’язаних із розробленими програмами. Частина цього захищеного процесу SDLC передбачає інтеграцію вимог безпеки в DevOps, результатом чого є підхід DevSecOps.

Зловмисники постійно шукають нові методи розгортання шкідливих програм і експлойтів. У випадку компрометації SolarWinds, згаданого у попередньому підрозділі, порушникам вдалося впровадити зловмисне програмне забезпечення в програму під час процесу створення, яке залишалося непоміченим, доки її не було розповсюджено серед тисяч клієнтів. Наслідки як для клієнтів, так і для репутації компанії були суттєвими, враховуючи, що SolarWinds використовували і приватні, і державні організації. Проте основними причинами масштабної кібератаки були відсутність процесу безпечної розробки та автоматизації перевірки змін в інформаційній системі SolarWinds.

SumoLogic, світовий лідер у сфері моніторингу SaaS, визначає шість критичних компонентів підходу DevSecOps. Першим є аналіз коду: вихідний код потрібно доставляти невеликими порціями для швидкого виявлення вразливостей. Наступним компонентом є процес керування змінами: учасники розробки повинні мати можливість надсилати та скасовувати зміни. Третім компонентом, який визначає компанія, є контроль відповідності: відповідність чинному законодавству та стандартам інформаційної безпеки повинна бути забезпечена. Після нього йде дослідження загроз: потенційні нові загрози повинні бути змодельовані з кожним оновленням коду або функцій програми. П'ятим компонентом є оцінка та виправлення вразливостей. Заключним компонентом є навчання з питань безпеки, яке повинно бути забезпечене для мінімізації ризиків, спричинених помилками людини.

Щоб гарантувати швидке виявлення загроз та проактивне реагування на них у підході DevSecOps, заходи безпеки можна об’єднати в безперервну інтеграцію та безперервне розгортання (CI/CD). Щоразу, коли створюють код, розробники запускають інструмент CI/CD, який виконує необхідні дії, як-от: закріплення коду в спільному репозиторії та сповіщення членів команди. Крім того, він може перевіряти такі фактори, як включення зовнішніх бібліотек, автентифікація, ліцензійні ризики, уразливості та випадковий обмін конфіденційною інформацією, як-от паролі/облікові дані у сховищі git. Сканування безпеки зображень контейнерів виконують перед тим, як вони надходять у CI/CD, і для цього доступні різні інструменти.

Типовий DevOps–процес включає такі етапи: план, надсилання коду, збірка, тестування, випуск та розгортання. У DevSecOps до кожної фази DevOps застосовують певні перевірки безпеки [36]. Дані перевірки безпеки перераховані у цьому списку:

1. План: на етапі планування виконують аналіз безпеки та створюють план для визначення сценаріїв того, як, де та коли проводитиметься тестування [36].
2. Надсилання коду: на цьому етапі потрібно регулярно проводити перевірку коду на наявність облікових даних користувачів [36].
3. Збірка коду: використання інструментів статичного тестування безпеки додатків (SAST) для відстеження вад коду перед його розгортанням у виробництві.
4. Розгортання: проведення динамічного тестування безпеки додатків (DAST).
5. Моніторинг: дослідження подій та інцидентів інформаційної безпеки, пов’язаних з додатком, шляхом збору системних подій [36].

Під час дослідження основних аспектів підходу DevSecOps було розроблено низку блок-схем для узагальнення методології перевірки додатків, що можуть слугувати основою для впровадження DevSecOps процесів у організацію при розробці інформаційної системи [36]. Зокрема, дані блок схеми зображено на Рис. 1.13 – 1.14 .

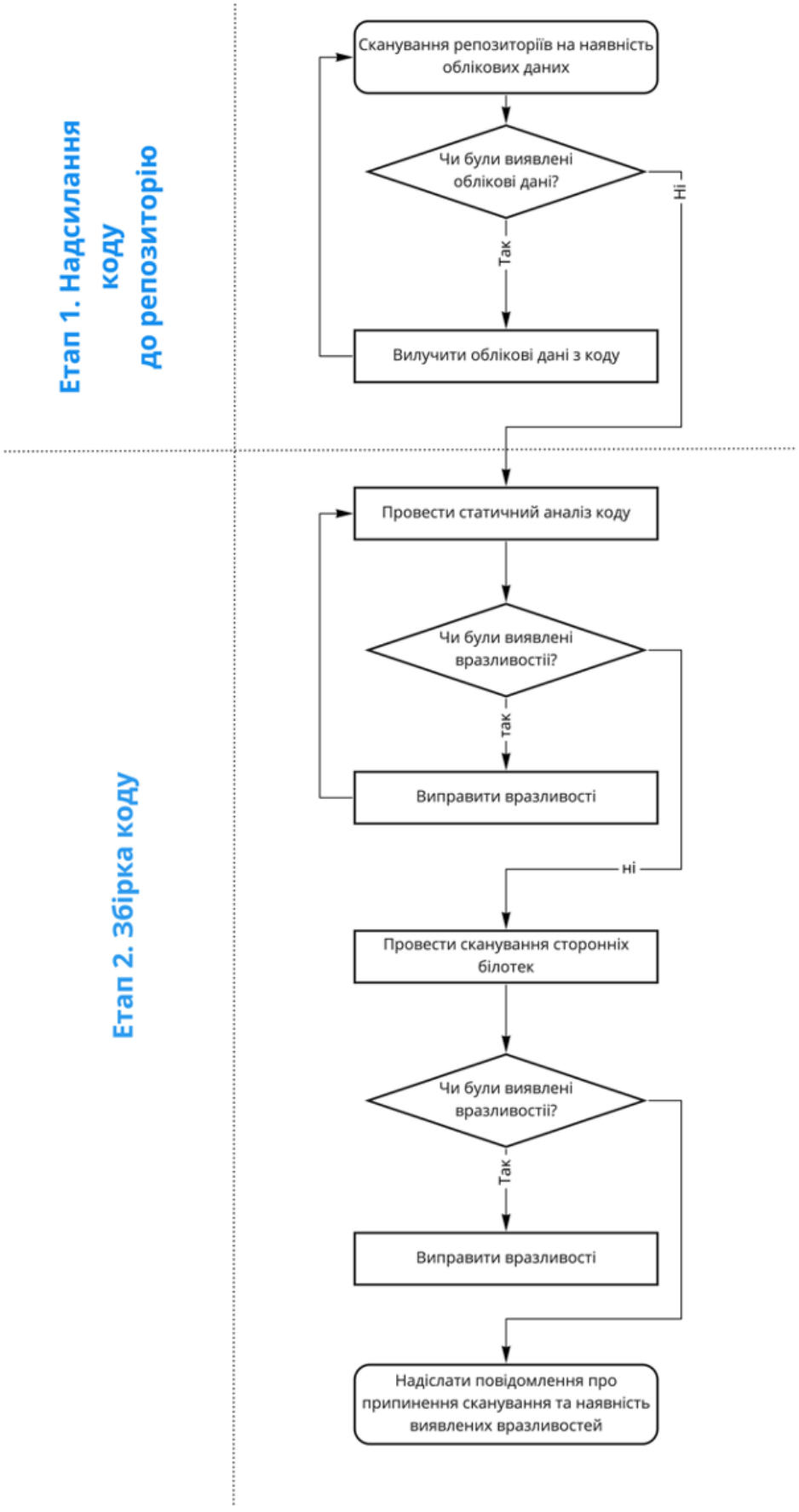



Рис. 1.6. Перевірка та збірка коду

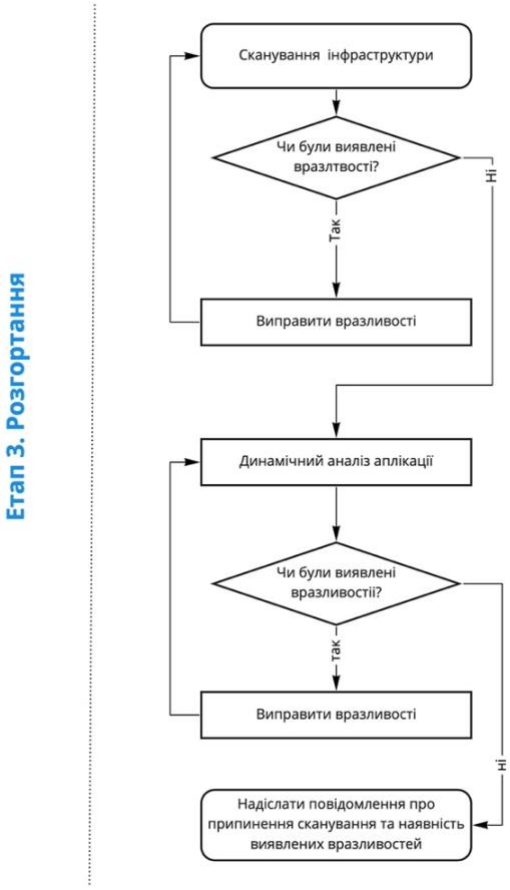



Рис.1.7. Етап розгортання

Під час реалізації процесу DevSecOps на етапах подання коду, створення та розгортання для організації вкрай важливо встановити критерії, які призупинять та заблокують процес змін, доки виявлені дефекти не будуть усунені. Зокрема, доцільно зупинити процес змін не лише для вирішення наявних проблем інформаційної безпеки, але й для таких дій, як розкриття персональних даних або паролів чи секретів у вихідному коді.

Моніторинг є важливим елементом у виявленні загроз у контексті DevSecOps. Використання єдиної системи для аналізу інцидентів, які стосуються програми та її основи, є критично важливим для глибокого аналізу. До систем, призначених для такого аналізу, можна зарахувати такі рішення, як Splunk, AlienVault OSSIM і Security Onion. Одним із основних завдань налаштування системи моніторингу є злиття інформації з усіх доступних джерел журналів в один моніторинговий хаб, що забезпечує повне охоплення подій, які виникають у інфраструктурі, додатках, системах контролю доступу до даних та брандмауера

веб-додатків.

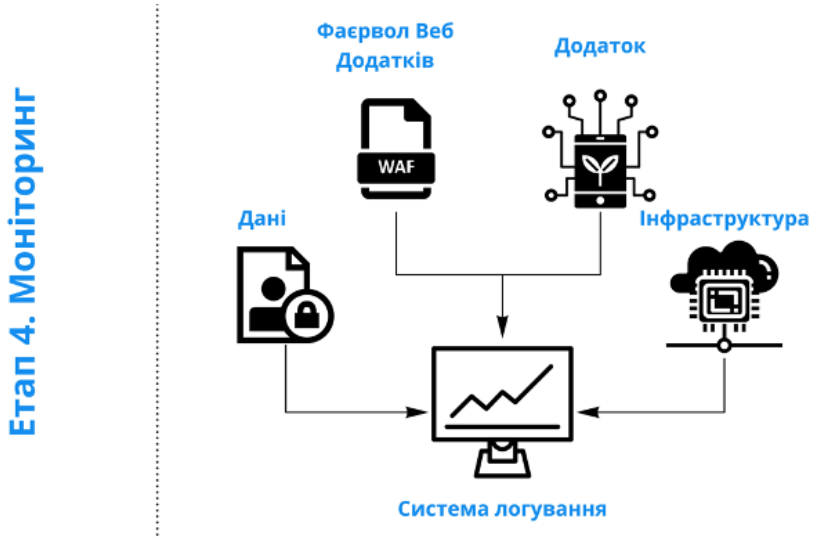



Рис.1.8. Моніторинг

Після інтеграції джерел подій до систем моніторингу наступним кроком є підєднання джерел дослідження загроз, до прикладу AlienVault OTX або IBM X-Force Xchange. Крім того, для автоматизації реагування на інциденти інформаційної безпеки вкрай важливо інтегрувати систему SIEM із системою SOAR (Security Orchestration, Automation, and Response). У рамках цього дослідження проаналізовано використання рішень для побудови DevSecOps процесу. До уваги брались рішення типу open-source. Дані рішення були зіставлені з контролями інформаційної безпеки стандартів ISO/IEC 27001:2022 та NIST CSF у різних організаціях та подано їх в Таблиці 1.3.

Таблиця 1.3.

Технічні рішення для побудови DevSecOps–процесу

| Контроль | Рішення | Ціль використання | ISO 27001: 2022 | NIST CSF |
|---|---|---|---|---|
| Оркестрація | Patrowl Demisto | Використовуються для автоматизації забезпечення інформаційної безпеки. | 6.8 | RS.CO-2 |

Продовження таблиці 1.3.

| | | | | |
|---|---|---|---|---|
| Статичне тестування безпеки додатків (SAST) | SonarQube,DerScanner | Інструменти статичного тестування безпеки додатків призначені для аналізу, щоб допомогти виявити недоліки безпеки. | 8.25 | PR.IP-2 |
| Динамічне тестування безпеки додатків (DAST) | OWASP ZAP, Arachni Nikto | Динамічне тестування безпеки додатків (DAST) досліджує програми на наявність вразливостей у розгорнутому середовищі. | 8.25 | PR.IP-2 |
| Аналіз складу програмного забезпечення SCA | Npm-audit, OWASP DC | Програмне забезпечення для аналізу складу програмного забезпечення (SCA) дозволяє користувачам керувати елементами відкритого коду своїх програм. | 8.25 | PR.IP-2 |
| Аудит конфігурацій хмарної інфраструктури | Scout-suite, вбудовані інструменти в хмарі | Інструмент перевірки безпеки, що дозволяє оцінювати контролі безпеки хмарної інфраструктури. | 5.36 | ID.RA-1 |

## 1.4 Дослідження можливостей використання штучного інтелекту

Сьогодні штучний інтелект (ШІ) став невід'ємною частиною більшості галузей промисловості. Зростання кібербезпеки в різних галузях була пов'язана зі списком вимог до ШІ, оскільки штучний інтелект дедалі більше впроваджується в бізнесі, сприяючи автоматизації [37]. Штучний інтелект швидко змінив численні галузі, започаткувавши нову еру ефективності та інновацій. На Рис. 1.71. представлені сектори застосування моделей ШІ.


Рис. 1.9. Сектори застосування моделей ШІ

Безумовно, штучний інтелект (ШІ) швидко змінив численні галузі, започаткувавши нову еру ефективності та інновацій. У секторі охорони здоров'я штучний інтелект робить значний прогрес. Від діагностики захворювань через медичну візуалізацію до прогнозування результатів пацієнтів ШІ покращує прийняття клінічних рішень і покращує догляд за пацієнтами. Крім того, фармацевтичні компанії використовують штучний інтелект для прискорення відкриття та розробки ліків, що потенційно революціонізує спосіб боротьби з хворобами.

У фінансовій галузі ШІ змінює ландшафт банківської справи та інвестицій. Ця тенденція була заснована на довгій історії штучного інтелекту у фінансах, і штучний інтелект нового покоління та машинне навчання фундаментально та плавно трансформують бачення, місії, цілі, парадигми, теорії, підходи, інструменти та соціальні аспекти економіки та стимулювання розумних фінансових технологій [38]. У фінансах безпека має першорядне значення. Банки використовують алгоритми на основі штучного інтелекту для виявлення шахрайства та моніторингу транзакцій у режимі реального часу. Системи виявлення шахрайських дій на основі штучного інтелекту можуть аналізувати дані транзакцій і виявляти аномалії, що вказують на шахрайські дії. Покращені методи автентифікації клієнтів завдяки штучному інтелекту зменшують ризик несанкціонованого доступу до облікових записів.

Такі компанії, як Tesla та Waymo, розробляють безпілотні автомобілі, що працюють на основі систем штучного інтелекту. Проте більшість програм штучного інтелекту зосереджено на розробці під'єднаних і автономних автомобілів, а не на оптимізації автомобільних операцій і виробничих процесів [39]. ШІ може захищати безпілотні автомобілі від кібератак, гарантуючи безпеку пасажирів і пішоходів. Безпека під'єднаних автомобілів викликає занепокоєння, і штучний інтелект може відігравати важливу роль у захисті автомобільних розважальних систем і мереж зв'язку.

У секторі роздрібної торгівлі штучний інтелект переосмислює досвід клієнтів. Інтернет-магазини використовують алгоритми рекомендацій, щоб

пропонувати продукти покупцям, збільшуючи продажі та задоволеність клієнтів. Звичайні магазини також використовують штучний інтелект для оптимізації управління запасами та підвищення безпеки за допомогою систем розпізнавання облич. Системи виявлення платіжного шахрайства на основі штучного інтелекту можуть визначати незвичні моделі платежів і викривати шахрайство з кредитними картками, захищаючи як клієнтів, так і компанії. Крім того, ШІ може допомогти контролювати запаси та запобігати крадіжкам.

У сільському господарстві ШІ допомагає фермерам оптимізувати врожайність і мінімізувати втрату ресурсів. Також у цій галузі відбувається швидка адаптація до ШІ у різних агротехнічних прийомах. Концепція когнітивних обчислень є тим, що імітує процес людського мислення як модель комп'ютер [40]. Рішення на основі штучного інтелекту аналізують дані датчиків, безпілотників і супутників, щоб надати інформацію про стан ґрунту, нашестя шкідників і погодні умови. Це дозволяє фермерам приймати обґрунтовані рішення щодо посадки, зрошення та боротьби зі шкідниками, зрештою підвищуючи продуктивність та стійкість сільського господарства. Безпека в сільському господарстві поширюється на ланцюг постачання. ШІ може відстежувати переміщення сільськогосподарської продукції, знижуючи ризик крадіжки або зараження, гарантуючи безпеку постачання продуктів харчування.

Потокові платформи можуть використовувати алгоритми ШІ, щоб рекомендувати вміст користувачам. На основі їхніх уподобань ШІ може виявляти та запобігати несанкціонованому розповсюдженню контенту, захищеного авторським правом, захищаючи інтелектуальну власність творців контенту. Крім того, ШІ допомагає потоковим службам та ігровим платформам захищати дані користувачів від зламу та кібератак.

Проведений аналіз застосування штучного інтелекту в різних галузях визначає його потенціал для зміни способів роботи компаній, взаємодії з клієнтами та вирішення складних завдань у сучасному світі. ШІ не тільки стимулює інновації, але й зміцнює безпеку, захищаючи основи цих галузей у цифровому та взаємопов’язаному ландшафті.

З огляду на те, що організації дедалі частіше використовують штучний інтелект для посилення захисту від кіберзагроз, зростає занепокоєння щодо конфіденційності даних, прозорості та підзвітності. Щоб вирішити ці проблеми та забезпечити відповідальне використання ШІ у сфері кібербезпеки, уряди та регулюючі органи в усьому світі активно розробляють законодавство для керування роботи ШІ.

## 1.5 Аналітичний огляд сучасного стану відповідальності за кіберзлочини згідно з Кримінальним кодексом України

Наведене у ч. 1 ст. 11 поняття злочину є формально-матеріальним. Із нього випливає, що злочином є діяння (дія або бездіяльність), якому властиві такі обов'язкові ознаки: 1) діяння, вчинене суб'єктом злочину; 2) воно є винним; 3) вказане діяння є суспільно небезпечним; 4) відповідне діяння передбачене чинним КК. Останнє, крім того, має на увазі, що обов'язковою ознакою злочину є також 5) кримінальна караність. Відсутність хоча б однієї з цих ознак вказує на відсутність злочину [41].

Термін “кіберзлочинність” виник в американському дискурсі на початку 1960-х років, що збіглося з відкриттям перших випадків злочинів, вчинених за допомогою комп’ютерних технологій. Першими “хакерами” були студенти Массачусетського технологічного інституту, які маніпулювали програмами нової комп’ютерної системи університету.

В Україні перші кроки у боротьбі з кіберзлочинністю були зроблені в 1994 році, коли були внесені зміни до Кримінального кодексу 1960 року. Ці зміни запровадили кримінальну відповідальність за зловмисне втручання в роботу автоматизованих систем, як це передбачено статтею 198–1 “Порушення роботи автоматизованих систем”. Це правопорушення стосується дій, що призводять до маніпулювання даними чи медіа, їх пошкодженням, або до розповсюдження програмного чи апаратного забезпечення, призначеного для незаконного вторгнення в автоматизовані мережі, здатного спотворити чи пошкодити дані чи медіа.

У “Доктрину інформаційної безпеки України” введено терміни “комп’ютерна злочинність” та “комп’ютерний тероризм”. Згодом до Стратегії національної безпеки, затвердженої Указом Президента України № 389/2012 від 8 червня 2012 року, були включені такі терміни, як кіберзлочинність, кіберзагроза та кібербезпека. Крім того, Закон України “Про основи національної безпеки України” також визнав поняття “комп’ютерний злочин” та “комп’ютерний тероризм”.

Закон України “Про основи забезпечення кібербезпеки України” дає визначення поняття “кіберзлочинність”. Він характеризує кіберзлочинність (також відому як комп’ютерна злочинність) як суспільно небезпечне злочинне діяння, що відбувається у віртуальному просторі та/або шляхом його використання. Це діяння тягне за собою кримінальну відповідальність, визначену законодавством України, або визнають злочин за міжнародними договорами, учасницею яких є Україна.

Незважаючи на те, що Закон України “Про кібербезпеку України” був зареєстрований 4 червня 2013 року, з метою вирішення термінологічних питань, він був остаточно відхилений, оскільки не містив термінології кіберзлочинності.

Згодом, 1 травня 2014 року, Президент України видав Указ № 449/2014, яким доручив розробити проект Стратегії кібербезпеки України та Закону “Про кібербезпеку в Україні”. Політика кібербезпеки України була офіційно затверджена Указом Президента № 96/2016 від 15 березня 2016 року.

Незважаючи на наявність відповідних законодавчих актів, вітчизняне законодавство лише частково відповідає сучасним вимогам. У ньому відсутні комплексні принципи розвитку інфраструктури інформаційної безпеки держави. Для подолання цього виклику вкрай необхідні законодавчі вдосконалення, які стануть основою єдиної державної політики, спрямованої на забезпечення інформаційного захисту та її ефективну реалізацію.

Огляд положень Кримінального кодексу України свідчить про значну диференціацію ступеня відповідальності за різні кримінальні правопорушення, про що детально зазначено в Таблиці 1.4.

Таблиця 1.4.

Положення Кримінального Кодексу України на предмет відповідальності за кіберзлочини

| Стаття КК України | Передбачена відповідальність |
|---|---|
| Стаття 361-1 | Несанкціоноване втручання в роботу електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку, що призвело до витоку, втрати, підробки, блокування інформації, спотворення процесу обробки інформації або до порушення встановленого порядку її маршрутизації [42]. |
| Стаття 361-2 | Ті самі дії, вчинені повторно або за попередньою змовою групою осіб, або якщо вони заподіяли значну шкоду [43]. |
| Стаття 361-1. ч.1 | Створення з метою використання, розповсюдження або збуту, а також розповсюдження або збут шкідливих програмних чи технічних засобів, призначених для несанкціонованого втручання в роботу електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку. [44]. |
| Стаття 361-1. ч.2 | Ті самі дії, вчинені повторно або за попередньою змовою групою осіб, або якщо вони заподіяли значну шкоду. [45]. |
| Стаття 361-2 ч. 1. | Несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або на носіях такої інформації, створеної та захищеної відповідно до чинного законодавства [46]. |
| Стаття 361-2 ч. 2. | Ті самі дії, вчинені повторно або за попередньою змовою групою осіб, або якщо вони заподіяли значну шкоду [47]. |

Продовження таблиці 1.4.

| | |
|---|---|
| Стаття 362 ч. 1. | Несанкціоновані зміни, знищення або блокування інформації, яка обробляється в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах чи комп'ютерних мережах або зберігається на носіях такої інформації, вчинені особою, яка має право доступу до неї [48]. |
| Стаття 362 ч. 2. | Несанкціоновані перехоплення або копіювання інформації, яка обробляється в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або зберігається на носіях такої інформації, якщо це призвело до її витоку, вчинені особою, яка має право доступу до такої інформації [49]. |
| Стаття 362 ч. 3. | Дії, передбачені частиною першою або другою цієї статті, вчинені повторно або за попередньою змовою групою осіб, або якщо вони заподіяли значну шкоду [50]. |
| Стаття 363 ч. 1. | Порушення правил експлуатації електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку або порядку чи правил захисту інформації, яка в них обробляється, якщо це заподіяло значну шкоду, вчинені особою, яка відповідає за їх експлуатацію [51]. |
| Стаття 363-1 ч. 1. | Умисне масове розповсюдження повідомлень електрозв'язку, здійснене без попередньої згоди адресатів, що призвело до порушення або припинення роботи електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку [52]. |
| Стаття 363-1 ч. 2. | Ті самі дії, вчинені повторно або за попередньою змовою групою осіб, якщо вони заподіяли значну шкоду [53]. |

Зважаючи на проведений огляд, можна виділити фактори, що сприяють комп'ютерним злочинам в Україні:

1. Важлива проблема виникає через відсутність узгодження законодавства міжнародним співтовариством. Дефіцит комплексного законодавчого регулювання в міжнародному масштабі є ключовим фактором кіберзлочинності. Це пояснюється тим, що вирішення всіх проблем, пов'язаних із захистом комп'ютерних технологій, не може бути ефективно досягнуто лише за допомогою національних законів окремих країн. Для вирішення цих проблем необхідні однакові підходи та однакові моделі поведінки. Цього можна досягти лише шляхом узгодження або гармонізації національного кримінального законодавства або додаткових міжнародних угод, призначених для переслідування правопорушників.

2. В Україні інформаційні відносини все ще перебувають у процесі розвитку. Наявне законодавство стосується переважно загальних питань, залишаючи деякі сфери недостатньо врегульованими, зокрема щодо захисту державної таємниці та інших категорій чутливої інформації.

Загалом положення Кримінального кодексу України спрямовані на забезпечення цілісності і безперервності роботи електронно-обчислювальних систем, а також захист інформації від несанкціонованого доступу та інших форм зловмисного втручання. Санкції включають штрафи, обмеження та позбавлення волі, і збільшуються у разі повторних правопорушень, дій групи осіб або заподіяння значної шкоди. Це відображає зростаючу глобальну тенденцію до посилення відповідальності за кіберзлочини, особливо з огляду на їх вплив на економіку.

**Висновки до 1 розділу**

У першому розділі дисертації проведено аналіз наукової літератури за темою дисертації, зокрема проаналізовано сучасні підходи до розслідування кіберзлочинів, вплив підходу DevSecOps, огляд сучасного стану відповідальності за кіберзлочини згідно з Кримінальним кодексом України. Проведене дослідження ддає підстави виділити наступне:

1. Проведений аналіз відмінностей у використанні різних типів інфраструктури інформаційних систем та їх загроз дав змогу зробити висновок, що єдиної моделі системи дослідження кібербезпеки на різних рівнях інформаційної системи не існує, тому актуальною є потреба у розробці моделі системи виявлення кібератак та дослідження кіберзлочинів на різних рівнях інфраструктури інформаційних систем.

2. Аналіз використання рішень ШІ встановив, що їх застосування є актуальним та необхідним для різних типів організацій та інфраструктури інформаційних систем. Однак проаналізована література дає змогу зрозуміти, що використання ШІ повинно бути контрольованим, і вимагає постійного вдосконалення та покращення впроваджених моделей штучного інтелекту.

3. Дослідження підходу DevSecOps визначило необхідність використання сканерів вразливостей на різних рівнях інфраструктури інформаційних систем, зокрема їх інтеграцію у процес розробки. Це забезпечить своєчасну реакцію на вразливості в інформаційних системах на етапі кодування та до впровадження змін у систему. Крім того, комплексний підхід до виявлення вразливостей може збільшити точність встановлення первинних причин кіберзлочинів.

Майбутні дослідження галузі кіберзлочинів повинні проводитися щодо практичної реалізації та використання моделей ШІ в рамках процесу DevSecOps, що дозволить точніше аналізувати події, можливі кіберзлочини та ймовірність їх виникнення.

# РОЗДІЛ 2. ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ ВЛАСТИВОСТЕЙ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЯВЛЕННЯ КІБЕРАТАК ТА АНАЛІЗУ КІБЕРЗЛОЧИНІВ

## 2.1. Проблематика сучасних рішень дослідження кіберзлочинів інформаційних систем

У цьому розділі розглянено проблематику сучасних методів дослідження кіберзлочинів на різних рівнях інфраструктури інформаційних систем та відмінності у системах захисту, залежно від типу інфраструктури.

У хмарних середовищах клієнти несуть відповідальність за контроль доступу, шифрування, моніторинг та механізми автентифікації для захисту своїх даних та застосунків. Водночас провайдери хмарних послуг гарантують безпеку базової інфраструктури. У випадку локальної інфраструктури контролі інформаційної безпеки містять заходи щодо фізичного доступу, антивірусні програми, брандмауери та керування вразливостями. Ці контролі є необхідними для захисту від шкідливого ПЗ, витоків даних та внутрішніх загроз.

Однак системи захисту без механізмів виявлення кібератак не можуть забезпечити надійний захист для хмарної, локальної чи гібридної інфраструктури. Сучасні стандарти, наприклад NIST, вводять концепцію "судової експертизи у хмарних обчисленнях" — це використання наукових принципів та методики для реконструкції минулих подій у хмарних обчисленнях за допомогою ідентифікації, збору, зберігання, аналізу та повідомлення про цифрові дані [54].

Для аналізу загроз інформаційної безпеки та кіберзлочинності використовують класичні рішення, як-от системи NIDS, HIDS, SIEM, EDR та хмарні системи дослідження подій.

Відмінність моніторингу хмарної інфраструктури зумовлено поширеністю взаємодії, керованої API, динамічного масштабування ресурсів і специфічних показників хмарних служб. Замість того щоб зосереджуватися винятково на мережевих підключеннях, важливо відстежувати виклики API, які лежать в основі хмарних служб. Тому, до звичних систем додано сервіс моніторингу хмарних рішень, який здебільшого надає постачальник хмарних послуг. Зокрема, до

найбільш популярних сервісів належать Azure Log Analytics та AWS GuardDuty. Проте класичні рішення дослідження кіберзлочинів також можуть використовуватись для хмарної інфраструктури. Для визначених особливостей кожного з рішень, їх недоліків та переваг запропонований огляд їх функціоналу та проблематики рішень для виявлення та аналізу кіберзлочинів.

IDS – це програмні або апаратні системи, які відстежують та аналізують хост-систему або мережу на наявність кібератак. IDS технологія повідомляє команди безпеки про факт атаки, тому відіграє важливу роль у захисті безпеки хмарних обчислень [55]. Те саме можна сказати й про важливість IDS систем для локальної інфраструктури, оскільки це базова технологія виявлення вторгнень. При використанні гібридного типу інфраструктури IDS варто використати для забезпечення безпеки сучасних бездротових мереж і систем, заснованих на їх інфраструктурі, враховуючи інтеграції внутрішньої інфраструктури у елементи мережі і зовнішнього управління інфраструктурою на основі хмарних платформ [56]. NIDS (система виявлення вторгнень у мережу) і HIDS (система виявлення вторгнень на хості) є двома важливими компонентами інфраструктури інформаційної системи. Вони працюють разом, щоб виявляти та запобігати різним типам вторгнень і загроз.

NIDS – це система виявлення вторгнень у мережі, яка спеціалізується на моніторингу мережевого трафіку для виявлення потенційних вторгнень і загроз. Система працює на рівні мережі, збираючи й аналізуючи пакети даних, щоб виявити підозрілі події, аномалії чи відомі сигнатури атак. Принцип роботи NIDS представлено схематично на Рис. 2.1.

На діаграмі зображеній на Рис. 2.1 вказано наступні класи:

1. Клас NIDS представляє саму систему виявлення вторгнень і включає атрибути ідентифікатора, назви опису, статусу і версійність. Ці атрибути описують NIDS.

2. Клас Мережевий Трафік представляє дані мережевого трафіку, які аналізує NIDS, включаючи такі атрибути, як вихідну IP, IP призначення, протокол і корисне навантаження.

3. Клас Повідомлення представляє сповіщення безпеки, створені NIDS у разі виявлення підозрілої активності. Він містить такі атрибути, як ідентифікатор, мітка часу, джерело, серйозність і опис.

4. Клас NIDS аналізує мережевий трафік, а саме дані мережевого трафіку. Клас NIDS генерує сповіщення, вказуючи, що він створює сповіщення безпеки.

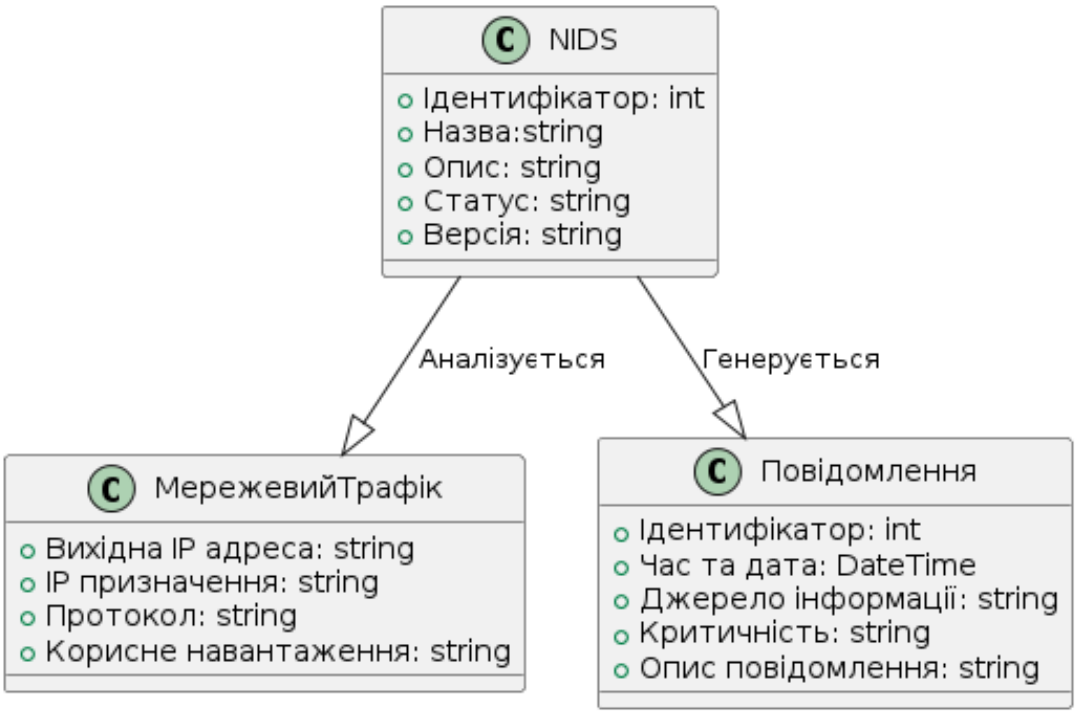



Рис. 2.1. Принцип роботи NIDS

HIDS – це система виявлення вторгнень на кінцевих точках, яка спеціалізується на виявленні вторгнень на хост-системах. Вони працюють на рівні хоста, постійно відстежуючи системні дії та конфігурації. Принцип роботи HIDS представлено схематично на Рис. 2.2.

Клас HIDS представляє сам HIDS і містить такі атрибути, як ідентифікатор, ім’я, опис, статус і версія, які описують HIDS. Клас "Хост" представляє хост-систему, яку контролює HIDS, і включає такі атрибути, як ім’я хоста, IP-адресу та операційну систему для опису хоста. Клас "Подія" представляє дані журналу, згенеровані хост-системою, включаючи такі атрибути, як id, час та дата, джерело, тип події та її опис. Клас "Повідомлення" представляє сповіщення безпеки, створені HIDS, коли він виявляє підозрілу активність. Він містить такі атрибути, як ідентифікатор, мітка часу, джерело, серйозність і опис. Клас HIDS слідкує за

хостом, тобто він стежить за системою хоста. Клас HIDS аналізує журнал, вказуючи, що він аналізує дані журналу з хосту. Клас HIDS генерує сповіщення, показуючи, що він генерує сповіщення безпеки на основі аналізу.

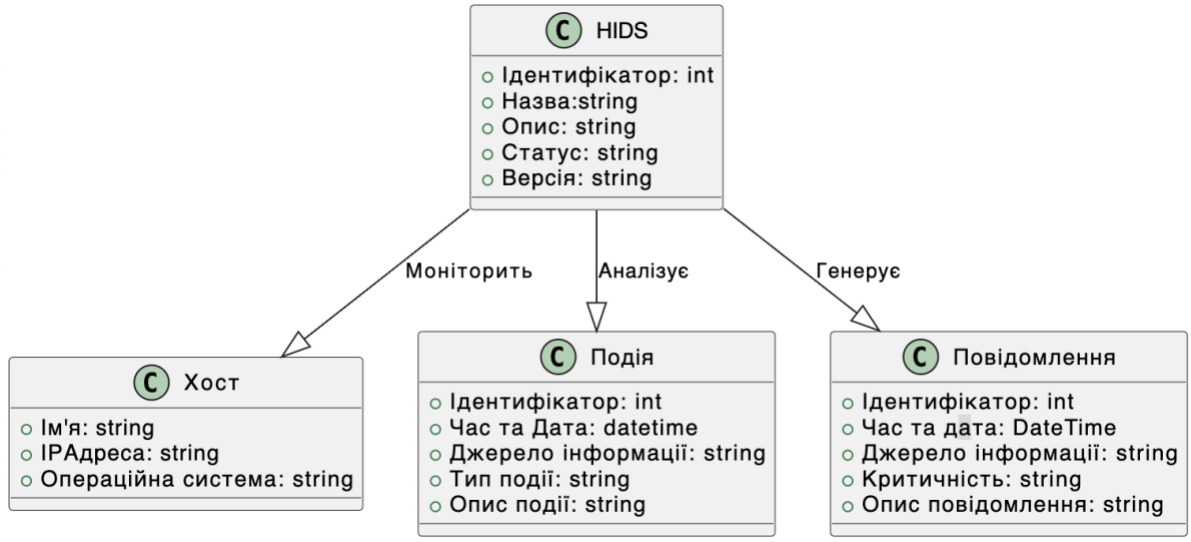



Рис. 2.2. Принцип роботи HIDS

SIEM (система безпеки та керування подіями) – це система, яка забезпечує централізовану обробку журналів шляхом збору журналів (здебільшого тих, що стосуються безпеки) з різних пристроїв і програм мережі, а також шляхом аналізу та зберігання цих журналів. Якщо система виявляє атаку, вона може реагувати через свої канали управління інцидентами, що включає сповіщення персоналу та ініціювання заходів протидії. SIEM також може допомогти організації дотримуватися правил збереження даних, де останнє може бути корисним у випадках електронного виявлення інцидентів (також відомого, як підготовка до судового розгляду) та криміналістики [57]. Варто зазначити, що наявність системи SIEM в додаток до NIDS і пропонує кілька ключових переваг. SIEM служить централізованою платформою для збору, кореляції та аналізу даних безпеки з багатьох джерел, включно з NIDS і HIDS. Це допомагає отримати повну картину стану безпеки організації, дозволяючи виявляти загрози в режимі реального часу, реагувати на кібератаки та проводити криміналістичний аналіз кіберзлочинів. Крім того, розширені аналітичні та кореляційні можливості SIEM допомагають командам безпеки визначати складні шаблони атак і ефективно визначати пріоритети попереджень, зменшуючи ризик помилкових спрацювань і втоми від

попереджень. Крім того, функції звітування та відповідності SIEM полегшують роботу з дотримання нормативних вимог і надають цінну інформацію для вдосконалення безпеки. Зрештою, поєднання SIEM з NIDS і HIDS підвищує загальну стійкість кібербезпеки організації, забезпечуючи класичний підхід до подолання кіберзагроз. SIEM–система схематично представлена на Рис. 2.3. Дане схематичне зображення визначає три прямокутні контейнери SIEM, джерела даних безпеки та зовнішні джерела даних. Усередині контейнера SIEM перераховані компоненти системи SIEM. Стрілки вказують на зв’язок між системою SIEM, джерелами даних безпеки і зовнішніми джерелами даних. Згідно з окресленими функціями SIEM–системи, визначено, що це самостійне рішення, яке може бути використане для виявлення кібератак. Проте зазвичай SIEM–системи не пропонують організаціям механізмів реагування на загрози чи інциденти інформаційної безпеки. Для таких цілей можна використовувати на мережевому рівні брандмауери нової генерації та системи виявлення вторгнень на кінцевих точках та реагування на них.

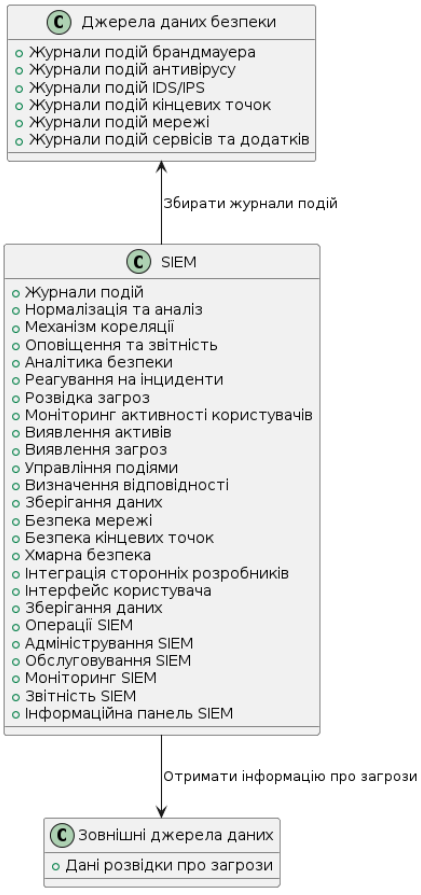



Рис. 2.3. Принцип роботи SIEM

Методи виявлення кібератак на кінцевих точках і реагування (EDR) призначені для подолання недоліків методів виявлення на основі поведінки. Оскільки EDR підтримують дії в різних ОС і розроблені з використанням відкритих джерел, вони дозволяють експертам у всьому світі співпрацювати та готувати відповіді швидше, ніж швидкість еволюції атаки [58]. Принцип роботи EDR представлено на Рис. 2.4, де визначено три прямокутні контейнери: EDR–система, кінцеві точки і центральний сервер. Контейнер EDR System представляє основні компоненти системи EDR. Контейнер кінцеві точки представляє кінцеві точки (пристрої), які відстежуються. Контейнер центральний сервер представляє центральний сервер, який керує агентами кінцевих точок та зберігає дані. На даній схемі EDR–система містить механізми виявлення вторгнень та на основі цих механізмів звітує про це користувачу системи.

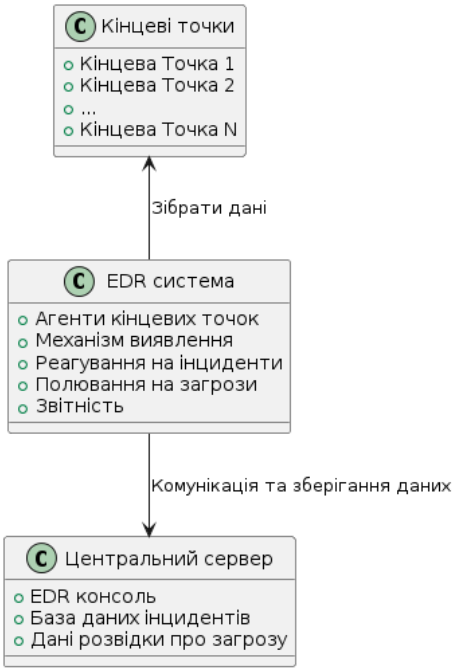



Рис. 2.4. Принцип роботи EDR

На схематичних зображеннях EDR та SIEM вказано використання даних розвідки про загрозу інформаційної безпеки. Інтеграція аналізу загроз у системи EDR і SIEM має важливе значення для випередження кіберзагроз, що розвиваються, підвищення точності виявлення загроз і забезпечення швидшого реагування на інциденти.

Розвідка про кіберзагрози проходить життєвий цикл, який називається життєвим циклом розвідки. Життєвий цикл розвідувальної інформації – це процес виявлення, збору та розробки необроблених даних та інформації, який перетворює на розвідувальну інформацію, яку використовують особи, які приймають рішення. Якщо цей процес виконувати правильно, діяльність розвідки може здійснюватись спрямовано та добре скоординовано, щоб задовольнити потреби користувачів. Дані розвідки збирають у систему дослідження загроз (систему збору даних розвідки про загрози) – це рішення кібербезпеки, призначене для збору, аналізу та поширення відповідної інформації про загрози та вразливості кібербезпеки. Ця система збирає дані з різних джерел, таких як розвідка з відкритим кодом, приватні канали та журнали внутрішньої безпеки, а потім обробляє ці дані, щоб надати організаціям практичну інформацію про потенційні загрози. Системи дослідження загроз допомагають командам безпеки адаптуватись до нових загроз, тактик і методів, які використовують кіберзлочинці. Ці системи відіграють вирішальну роль у покращенні стану кібербезпеки організації, надаючи своєчасну та контекстну інформацію, яка допомагає виявляти загрози, реагувати на інциденти та приймати рішення. Схематичне зображення системи дослідження загроз наведено на Рис.2.5. У ньому визначено два прямокутні контейнери: система розвідки про загрози і зовнішні джерела даних. Контейнер система розвідки про загрози представляє основні компоненти системи аналізу загроз. Контейнер зовнішні джерела даних представляє зовнішні джерела даних про загрози, наприклад: загальнодоступні та приватні канали загроз і постачальників даних.

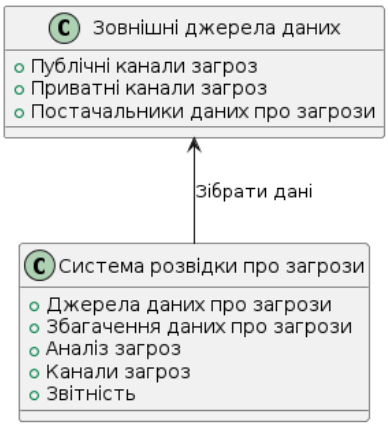



Рис. 2.5. Схематичне зображення системи дослідження загроз

Найбільш поширеним засобом для виконання статистичного аналізу спроб реалізації різних інформаційних загроз є найбільша з існуючих розподілених IP з відкритою архітектурою - Інтернет, який функціонує на основі стеку протоколів TCP/IP. Аналіз полегшується наявністю інформації про результати моніторингу інцидентів безпеки, опубліковані різними організаціями, наприклад, CERT, CIAC, NIPC [59].

Як зазначалось вище, для аналізу кіберзлочинів можна використовувати також хмарні системи, такі як Amazon GuardDuty та Azure Log Analytics. Вони пропонують розширені можливості для моніторингу та аналізу журналів, подій і мережевого трафіку та API викликів, що дозволяє організаціям завчасно виявляти та досліджувати кіберзагрози в хмарній інфраструктурі [60]. Схематичне зображення принципу роботи таких систем наведено на Рис. 2.6.


Рис. 2.6. Схематичне зображення системи дослідження кіберзлочинів в хмарі

Схематичне зображення визначає три прямокутні контейнери: система дослідження, хмарне середовище і сторонні джерела даних. У контейнері система дослідження перераховані типові компоненти спеціальної хмарної системи дослідження кіберзлочинів. Стрілки вказують на зв’язок між системою дослідження та хмарним середовищем (зібрані дані) і зовнішніми джерелами даних (імпортовані журнали та дані про загрози). Завдяки інтеграції спеціалізованих хмарних систем у свої стратегії безпеки організації можуть підтримувати надійну безпеку, дотримуватися вимог міжнародних стандартів та зменшувати ризики витоку даних.

Важливість інтеграції різних систем кібербезпеки, таких як EDR, SIEM, системи дослідження загроз та NIDS/HIDS для аналізу кіберзлочинів для складових інфраструктури інформаційних систем полягає в їхній спільній здатності створювати комплексну багатошарову стратегію захисту від кіберзлочинів. Ці системи працюють разом, дозволяючи організаціям виявляти та реагувати на кіберзагрози. EDR забезпечує видимість діяльності кінцевих точок у режимі реального часу, SIEM збирає та корелює дані з усієї організації, системи дослідження загроз забезпечують своєчасний контекст загроз, а NIDS/HIDS відстежує поведінку мережі та хостів. Разом вони пропонують цілісне виявлення загроз, можливості раннього виявлення кібератак, швидке реагування та прийняття обґрунтованих рішень під час дослідження кіберзлочинів. Цей інтегрований підхід посилює стійкість організації проти кіберзлочинців, зменшуючи вразливість, зводячи до мінімуму збитки та захищаючи критичні активи. Сучасні системи безпеки пропонують організаціям можливості для виявлення загроз безпеці та реагування на них. Однак вони також мають власний набір викликів і проблем:

1. Залежно від продукту та типу інсталяції (локальна чи хмарна) вартість може відрізнятися, та комерційні рішення часто пропонують високу вартість SIEM–рішень [60].
2. Системи SIEM збирають і аналізують велику кількість даних безпеки з різних джерел. Таке перевантаження даними може призвести до великої кількості сповіщень, через що командам безпеки буде складно визначити пріоритети та

ефективно реагувати. Компанії повинні прагнути до впровадження SIEM, які, аналізуючи сотні мільйонів файлів журналів подій, визначатимуть 2-3 сповіщення, які можна вжити, із контекстом. SIEM без будь-якої оптимізації буде створювати 1000 сповіщень [61].

3. Рішення EDR створюють значний обсяг даних кінцевої точки, включно з журналами, подіями та телеметрією. Аналіз цих даних може потребувати ресурсів і генерувати велику кількість сповіщень. Системи аналізу загроз можуть надати велику кількість даних про нові загрози та вразливості, проте організаціям може бути важко відфільтрувати та ефективно застосувати ці дані до свого середовища.

4. Враховуючи заявлену раніше кількість досліджень, які зможе виконати середній аналітик SOC, стає зрозуміло, що компанії зі 100 мільйонами файлів журналів на день знадобиться приблизно 100 аналітиків даних [62]. Велика кількість сповіщень від систем SIEM може призвести до втоми від сповіщень, коли аналітиків безпеки перевантажують великою кількістю сповіщень, що призводить до пропуску або затримки відповідей на критичні загрози. Рішення EDR можуть генерувати численні сповіщення, багато з яких можуть бути помилковими. Це може призвести до того, що служби безпеки втратять чутливість до сповіщень і пропустять справжні загрози.

5. SIEM–системи можуть генерувати хибно–позитивні сповіщення через неправильну конфігурацію правил або неадекватну кореляцію. Це витрачає час і ресурси на дослідження неіснуючих загроз. Якщо брати до уваги EDR рішення, то вони можуть ініціювати сповіщення на основі підозрілої поведінки, яка іноді може бути нормальною діяльністю.

6. SIEM–системи та EDR–системи часто обмежені у типі подій, які вони можуть досліджувати, та вони часто не підтримують усі типи джерел подій та сервісів, що можуть надсилати дані до SIEM. Інтеграція каналів аналізу загроз в існуючі системи безпеки може бути складною та трудомісткою. Це вимагає ретельного налаштування, щоб забезпечити ефективне використання відповідних даних про загрози. Також без процесу виявлення вразливостей наявність системи аналізу загроз може бути недоцільним, оскільки виявлення вразливостей відіграє

ключову роль у розробці інформаційних систем. Однак існує багато різних інструментів і методів виявлення вразливостей на вибір, і недостатньо інформації про те, які методи виявлення вразливостей використовувати та коли [63].

7. Керування безпекою в хмарній інфраструктурі часто передбачає використання кількох хмарних провайдерів та інструментів. Інтеграція цих різнорідних систем може вимагати додаткових ресурсів.

8. Суб'єкти загрози постійно адаптуються та змінюють тактику, що ускладнює для організацій можливість не відставати від нових загроз. Канали розвідки про загрози необхідно постійно оновлювати та вдосконалювати. Хмарні середовища є привабливими цілями для зловмисників і продовжують з'являтися все нові вектори атак. Організації повинні постійно адаптовувати свої заходи безпеки.

Підсумовуючи, варто зауважити, що хоча сучасні системи безпеки пропонують потужні можливості, вони також створюють проблеми, пов'язані з керуванням даними, надлишком сповіщень, складністю інтеграції, конфіденційністю, відповідністю та постійно змінним ландшафтом загроз. Організації повинні вирішувати ці виклики, щоб максимізувати ефективність своїх операцій безпеки та захисту від нових загроз. Для покращення процесу аналізу загроз та їх виявлення, (це стосується саме інформаційної системи), можна впровадити процес DevSecOps для постійного тестування безпеки, та варто зазначити, що одним із найважливіших атрибутів будь-якого тестування безпеки є покриття. Щоб оцінити безпеку інформаційної системи, автоматизований сканер повинен мати можливість точно інтерпретувати програму .

Для покращення процесів дослідження кіберзлочинів рішення типу SIEM–систем, EDR та системи аналізу безпеки хмарних рішень можуть використовувати алгоритми машинного навчання. Методи штучного інтелекту, засновані на моделюванні аналізу безпеки, можна використовувати для вирішення різних проблем і завдань кібербезпеки, таких як автоматична ідентифікація зловмисних дій, виявлення фішингу, виявлення зловмисного програмного забезпечення, прогнозування кібератак, виявлення шахрайства, управління контролем доступу, виявлення аномалій або вторгнень тощо [64]. Відповідний аналіз можливостей

алгоритмів штучного інтелекту у сфері дослідження кіберзлочинів описано у наступному розділі.

## 2.2. Використання алгоритмів штучного інтелекту для дослідження подій та інцидентів інформаційної безпеки

Аномалія – це неправомірна точка даних, згенерована процесом, що відрізняється відмінним від того, що згенерувало решту даних [65]. ШІ – це нова технологічна наука, яка вивчає та розробляє теорії, методи, техніки та програми, які симулюють, розширюють людський інтелект [66]. Алгоритми машинного навчання (ML) – це гілка штучного інтелекту, яка тісно пов’язана з обчислювальною статистикою (і часто з нею збігається), яка також зосереджена на створенні прогнозів. ML має тісні зв’язки з математичною оптимізацією, яка надає методи, теорію та області застосування ML часто використовують у сфері кібербезпеки. Для прикладу у сфері застосування NFC-технологій завдяки виявленню аномалій, покращенню біометричної автентифікації, захисту від типових атак: "людина посередині" (MITM), оптимізації шифрування та токенізації. Адже NFC бездротова технологія, то виникнення інциденту прослуховування каналу комунікації є ймовірним [67]. Алгоритми машинного навчання можуть аналізувати трафік на предмет аномалій, що вказують на можливу атаку для виявлення пристроїв, які намагаються санкціонувати комунікацію між NFC-пристроями. Також, їх можна використовувати, щоб провести аналіз геолокаційних даних для виявлення аномальних переміщень, що можуть свідчити про спробу реплікації профілю користувача, що є критично важливим завданням у сучасному світі. Адже за успішного виконання цієї атаки зловмисник може створити копію цифрового профілю користувача і використовувати її для несанкціонованого доступу до сервісів, фінансових рахунків або інших ресурсів, пов'язаних з цим профілем [68].

Отож, одним із ключових застосувань алгоритмів ML у безпеці є виявлення аномалій. Моделі ML можна навчити виявляти нормальну модель поведінки в інформаційній системі. Будь-які відхилення від нормальної моделі поведінки позначаються як аномалії. Наприклад, використання незвичайних запитів, стрибки

мережевого трафіку або несподіване споживання ресурсів можуть ініціювати сповіщення. Алгоритми ML також можуть класифікувати події безпеки, як доброякісні або зловмисні. Вони можуть аналізувати різні джерела даних, наприклад, журнали безпеки, мережевий трафік або поведінку користувачів, щоб визначити характер подій. Наприклад, моделі ML можуть розрізняти законні електронні листи та спроби фішингу на основі аналізу вмісту листа, поведінки відправника та історичних даних про загрози.

Одним із популярних алгоритмів машинного навчання, який використовують в кібербезпеці, є ізоляційний ліс (Isolation Forest) завдяки здатності до виявлення аномалій. Ізоляційний ліс не використовує відстань або щільність для виявлення аномалій, уникаючи великих обчислень на основі методів відстані та щільності [65]. Його ефективність і масштабованість роблять його добре придатним для обробки великих обсягів журнальних даних, що є важливим аспектом своєчасного виявлення загроз, адже згідно з останніми дослідженнями IBM, виявлення інцидентів інформаційної безпеки становить 208 днів [69]. На відміну від деяких інших алгоритмів, ізоляційний ліс менш чутливий до основного розподілу даних, що робить його адаптованим до різноманітних і змінюваних моделей атак, які спостерігаються в кібербезпеці. Крім того, це однокласовий алгоритм навчання, що означає, що він може вивчати нормальні характеристики подій, не вимагаючи позначених аномалій під час навчання.

Ще одним видом моделей, яке часто використовують спеціалісти інформаційної безпеки, є глибоке навчання, що використовує нейронні мережі з кількома рівнями аналізу та інтерпретації складних даних. У контексті кібербезпеки моделі глибокого навчання довели свою високу ефективність у вирішенні складних і різноманітних проблем безпеки [70]. Вони особливо добре підходять для завдань безпеки на основі зображень, аналізу мережевого трафіку та обробки даних часових рядів. Одним із основних застосувань глибокого навчання в безпеці є використання загорткових нейронних мереж (CNN) для завдань безпеки на основі зображень. CNN вправно аналізують зображення та відео, щоб виявити потенційні загрози безпеці. Наприклад, їх можна використовувати в системах

спостереження для розпізнавання несанкціонованих осіб або об’єктів у зонах обмеженого доступу, посилюючи фізичну безпеку [71]. На відміну від CNN, повторювані нейронні мережі (RNN), тип моделі глибокого навчання є цінними для аналізу даних на основі послідовності. Вони використовуються в безпеці для таких завдань, як аналіз мережевого трафіку, де вони можуть ідентифікувати шаблони в потоках даних, які можуть вказувати на порушення безпеки або підозрілу діяльність. RNN також ефективні для аналізу даних часових рядів для виявлення аномальної поведінки [72].

Моделі глибокого навчання, включаючи CNN і RNN, використовують в системах виявлення вторгнень у реальному часі. Ці системи відстежують мережевий трафік, виявляючи шаблони, пов’язані з відомими та новими загрозами. Вони можуть швидко надсилати сповіщення та запускати автоматичні відповіді для пом’якшення атак, наприклад блокування шкідливих IP-адрес або розміщення скомпрометованих пристроїв на карантин. Глибоке навчання часто використовується в антивірусних рішеннях для покращення виявлення зловмисного програмного забезпечення. Також можуть бути використані для аналізу поведінки користувачів і об’єктів (UEBA), щоб відстежувати поведінку користувачів і об’єктів у мережах. Вони виявляють відхилення від встановлених норм, сигналізуючи про потенційні внутрішні загрози або скомпрометовані облікові записи. Отже, Deep Learning є важливим інструментом у сфері кібербезпеки, зокрема в подіях безпеки та розслідуванні інцидентів. Хоча глибоке навчання досягло величезного успіху в перетворенні багатьох завдань інтелектуального аналізу даних і машинного навчання, але популярні методи глибокого навчання непридатні для виявлення аномалій через деякі унікальні характеристики аномалій, наприклад: рідкість, неоднорідність та непомірно високу вартість збору великих даних. Також було виявлено проблеми конфіденційності та безпеки DL, що модель DL можна вкрасти або провести зворотний інженерний аналіз, можна отримати конфіденційні навчальні дані, навіть можна відновити впізнаване зображення обличчя жертви. Крім того, нещодавні роботи виявили, що

модель DL є вразливою до змагальних прикладів, порушених непомітним шумом, який може призвести модель DL до неправильного прогнозування [72].

Випадковий ліс (Random Forest) – це універсальний алгоритм машинного навчання, що відіграє ключову роль в аналізі інцидентів і подій у сфері кібербезпеки. Модель випадкового лісу фактично є процесом навчання для класифікації, регресії та інших завдань. Точніше кажучи, RF-модель базується на деревах рішень і механізмі пакетування (тобто завантажувального агрегування), щоб уникнути проблеми переобладнання складних дерев рішень [73]. Його здатність обробляти структуровані та неструктуровані дані в поєднанні з ефективністю у виявленні закономірностей і аномалій робить його цінним активом для зміцнення цифрового захисту та реагування на інциденти безпеки. Вказана модель добре справляється з виявленням аномалій у даних подіях. Вивчаючи зразки історичних даних, він може розпізнавати відхилення від норми. Під час інциденту безпеки не всі сповіщення повинні аналізуватись однаково, і алгоритм випадкового лісу може визначати пріоритет сповіщень на основі їх серйозності та потенційного впливу. Призначаючи оцінки ризику подіям, це допомагає командам безпеки, які займаються реагуванням на інциденти, зосередити свої зусилля в першу чергу на найбільш критичних загрозах. Також модель може аналізувати поведінку користувачів і об’єктів у мережі чи системі. Випадковий ліс може виявляти незвичайні дії користувача, такі як повторні невдалі спроби входу або неавторизований доступ до конфіденційних ресурсів. Цей аналіз поведінки допомагає точно визначити внутрішні загрози та зовнішні атаки.

Тому модель випадкового лісу можна використовувати в аналізі інцидентів і подій безпеки, що дає змогу організаціям виявляти, оцінювати та реагувати на інциденти безпеки.

Для визначення алгоритму, який найкраще підходить для виявлення аномалій інформаційної безпеки, було проведено тестування моделей між собою. Для порівняння було проведено наступні експерименти:

1. Створено три моделі виявлення аномалій: випадковий ліс та ізоляційний ліс написані на Python із використанням таких бібліотек, як Pandas для обробки даних

та Scikit-learn для завдань машинного навчання, що забезпечує реалізацію алгоритму Random Forest та інструментів для оцінки моделі. Модель Deep Learning була розроблена для виявлення аномалій за допомогою багаторівневого класифікатора персептрона (MLP), типу нейронної мережі, реалізованої в бібліотеці scikit-learn.

2. Створено два довільних набори даних для навчання трьох моделей. Перший – це довільний набір даних. Другий – це довільний набір подій з вебсерверу NGINX з атрибутами атаки сканування.

3. Кожна модель протестована на однаковому наборі даних: журналу подій з атрибутами атаки сканування.

4. Для оцінки моделі використовували наступні показники: акуратність, влучність, відкликання, оцінка F1 та площа під кривою ROC (AUC-ROC) [74]. Акуратність – це співвідношення правильно визначених передбачень до загальної кількості передбачень[75], розрахована дана метрика наступним чином:

$$\text{Акуратність} = \frac{\text{TP} + \text{TN}}{\text{TP+TN+FP+FN}}, \tag{2.1}$$

де TP – точно позитивні , TN – точно негативні, FP – хибно позитивні та FN – хибно негативні результати.

Влучність – відношення точно позитивних до суми хибних та точно позитивних результатів [76]. Метрика описується наступною формулою:

$$\text{Влучність} = \frac{\text{TP}}{\text{TP+FP}}, \tag{2.2}$$

Відкликання – це метрика, що вказує на здатність моделі знаходити всі відповідні випадки (позитивні) у наборі даних. Дана метрика розраховується так:

$$\text{Відкликання} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.3}$$

Оцінка F1 є середнім гармонійним значенням влучності та відкликання. Це спосіб об’єднати обидва показники в єдину оцінку, яка фіксує як помилкові позитивні, так і помилкові негативні результати. Дана метрика розраховується такою формулою:

$$\text{F1 Оцінка} = 2 \times \frac{\text{Влучність} \times \text{Відкликання}}{\text{Влучність} + \text{Відкликання}}, \quad (2.4)$$

Також важливою метрикою є площа під кривою ROC (AUC-ROC). Крива AUC-ROC є вимірюванням продуктивності для проблем класифікації за різних порогових значень. ROC — це крива ймовірності, а AUC — ступінь або міра роздільності. AUC вказує, наскільки модель здатна розрізняти класи. Чим вищий AUC, тим краще модель прогнозує 0 класів як 0 і 1 клас як 1. Оцінка AUC 0,5 свідчить про відсутність випадкового вгадування. Хоча AUC-ROC є графічним представленням і не має простої формули, загальна ідея полягає в тому, щоб обчислити площу під кривою ROC, яка є графіком істинного позитивного результату (TPR) проти помилкового позитивного показника (FPR) при різних порогових значеннях.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2.5)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad (2.6)$$

Результати тестування моделей зображені на Рис. 2.7.

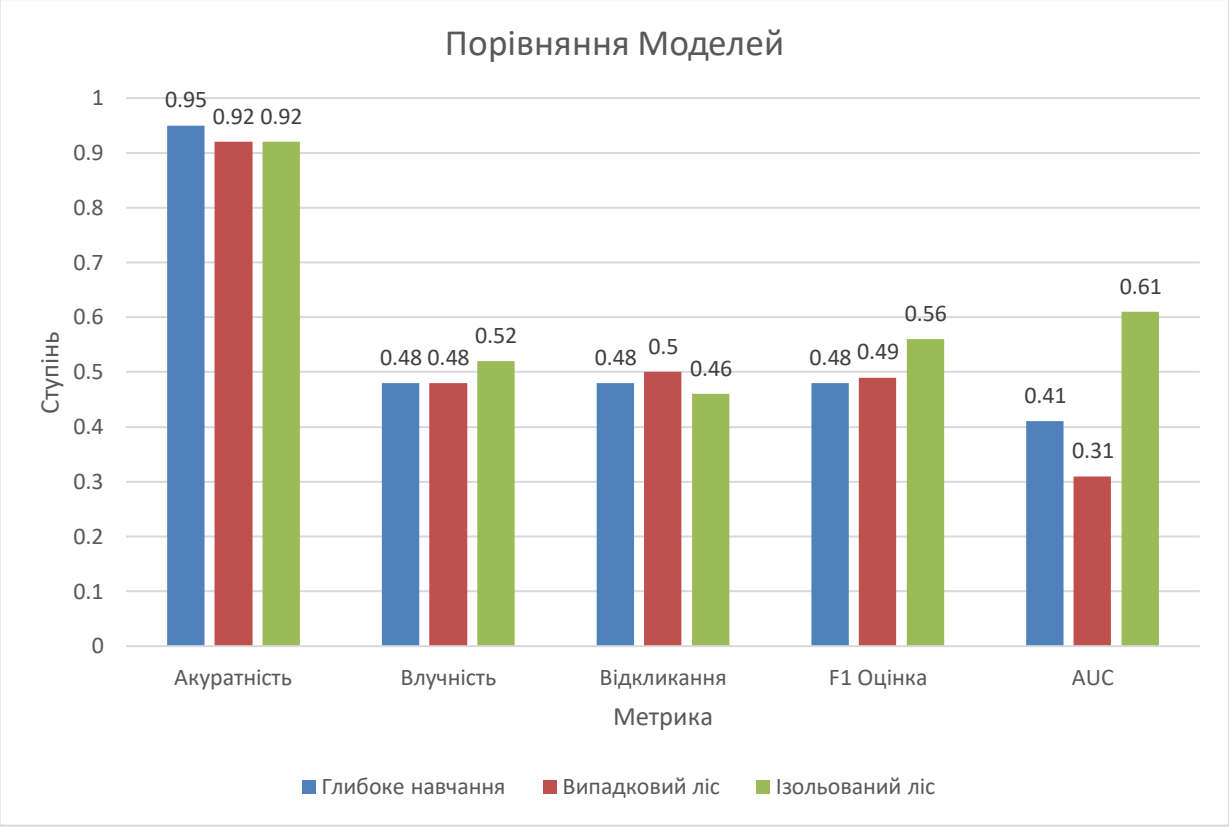



Рис. 2.7. Порівняння моделей

За результатами даного експерименту визначено, що під час виявлення аномалій з набору даних подій інформаційної безпеки згенерованих атакою сканування на сервіс NGINX, модель глибокого навчання показує помірні показники з найнижчою AUC, що вказує на проблеми з розрізненням класів. Модель випадкового лісу демонструє високу акуратність, але її влучність та оцінка F1 нижчі порівняно з іншими моделями, а показник AUC найгірший серед трьох моделей. Це говорить про труднощі в ефективному виявленні аномалій.

Модель ізоляційного лісу відображається з високою акуратністю та влучністю та найвищим показником AUC та F1, що свідчить про найнижчий рівень хибних спрацювань.

Також у рамках порівняння моделей було проведено аналіз літератури за тематикою дослідження алгоритмів машинного навчання та сформовано таблицю 2.1, яка використовує стандартні для алгоритмів машинного навчання критерії для формування порівняльної характеристики: складність, здатність працювати з різними типами даних, виявляти аномалії, інтерпретувати результати, розмір навчальних даних, масштабованість, обробка даних у реальному часі, наявність помилкових спрацювань та діапазон застосування.

Таблиця 2.1.

Порівняльна характеристика алгоритмів машинного навчання для аналізу аномалій

| Критерій | Глибоке навчання | Алгоритм ізоляційний ліс | Алгоритм випадковий ліс |
|---|---|---|---|
| Складність | Глибокі та складні нейронні мережі з багатьма параметрами. | Простий і ефективний алгоритм на основі дерев рішень. | Ансамбль дерев рішень, що пропонують помірну складність. |

Продовження таблиці 2.1.

| | | | |
|---|---|---|---|
| Здатність працювати з різними типами даних | Ефективно для неструктурованих даних. | Підходить для структурованих і неструктурованих даних, у тому числі табличних і журнальних даних. | Універсальний для структурованих даних, аналізу журналів і даних подій. |
| Здатність виявляти аномалії | Ефективний для виявлення аномалій на основі послідовності, але може потребувати великих наборів даних. | Здатний виявляти аномалії, особливо у великих наборах даних. | Здатний виявляти аномалії, але може потребувати розробки функцій. |
| Здатність інтерпретувати результати | Часто вважається “чорним ящиком” з обмеженою інтерпретацією для прийняття рішень. | Забезпечує певну інтерпретацію через оцінку важливості функції. | Пропонує помірну інтерпретацію завдяки аналізу важливості ознак. |
| Розмір навчальних даних | Для ефективного навчання потрібні великі обсяги позначених даних, які можуть бути складними для отримання в галузі кібербезпеки. | Для навчання потрібна менша кількість позначених даних. | Для навчання зазвичай достатньо помірних обсягів позначених даних. |

Продовження таблиці 2.1.

| | | | |
|---|---|---|---|
| Масштабованість | Може потребувати інтенсивних обчислень, особливо для складних архітектур. | Масштабований і ефективний для великих наборів даних. | Масштабується, але ефективність може знизитися з дуже великими наборами даних. |
| Обробка в реальному часі | Може бути складним для обробки в реальному часі через обчислювальні вимоги. | Добре підходить для обробки в реальному часі та швидкого виявлення аномалій. | У багатьох випадках підходить для обробки в режимі реального часу або майже в реальному часі. |
| Наявність помилкових спрацьовувань | Схильний до більшої кількості помилкових спрацьовувань через складність. | Відомий своєю здатністю зменшувати помилкові спрацьовування. | Пропонує помірну частоту хибних спрацьовувань залежно від налаштування. |
| Діапазон застосування | Ефективний для аналізу зображень, обробки природної мови та розпізнавання складних образів. | Дуже добре підходить для виявлення аномалій у структурованих і неструктурованих даних. | Універсальний для широкого спектру завдань кібербезпеки, включаючи виявлення атак і аналіз журналів подій. |

Отож, зважаючи на наведений аналіз, модель ізоляційного лісу пропонує наступні переваги над випадковим лісом і алгоритмами глибокого навчання для виявлення аномалій, що може бути основою для аналізу подій та інцидентів інформаційної безпеки:

1. Ефективність і швидкість: ізоляційний ліс може ефективно обробляти великі обсяги даних і швидко виявляти аномалії.

2. Масштабованість: ізоляційний ліс має високу масштабованість і здатність обробляти дані великого розміру.

3. Надійність до розподілу даних: ізоляційний ліс менш чутливий до основного розподілу даних. Він добре працює як з перекошеним, так і з мультимодальним розподілом даних, що робить його адаптованим до різних наборів даних кібербезпеки. Навпаки, методи випадкового лісу і глибокого навчання можуть потребувати додаткової попередньої обробки даних і налаштування для ефективної обробки різноманітних розподілів даних.

4. Однокласове навчання: Ізоляційний Ліс – це однокласовий алгоритм навчання, тобто він може вивчати характеристики звичайних даних без потреби в позначених аномаліях під час навчання. Це робить його придатним для аналізу нових моделей атак.

5. Простота впровадження: ізоляційний ліс відносно простий у реалізації та не потребує значного налаштування гіперпараметрів, що робить його привабливим вибором для швидкого створення прототипів і розгортання.

Хоча ізоляційний ліс пропонує ці переваги для багатьох сценаріїв виявлення аномалій у сфері кібербезпеки, важливо зазначити, що вибір алгоритму завжди має ґрунтуватися на конкретних вимогах і характеристиках даних, а також на поточному завданні кібербезпеки. У деяких випадках методи Random Forest або Deep Learning можуть бути більш придатними, особливо для завдань, які включають складні шаблони, аналіз зображень або обробку природної мови. Тому вибір найкращого алгоритму залежить від конкретного контексту та цілей завдань кібербезпеки.

## 2.3. Порівняльна характеристика застосування класичних рішень та моделей ШІ для дослідження кіберзлочинів

Індустрія безпеки успішно застосувала деякі методи штучного інтелекту. Використання варіюється від пом'якшення впливу атак типу "відмова в обслуговуванні", криміналістики, систем виявлення вторгнень, внутрішньої безпеки, захисту критичної інфраструктури, витоку конфіденційної інформації, контролю доступу та виявлення шкідливих програм [76]. Розвиток машинного навчання також значно підвищив операційну ефективність і результативність.

В інформаційній системі, що розвивається, важливість застосунків виявлення вторгнень і керування подіями постійно зростає, оскільки вони стали незамінними інструментами для організацій, щоб моніторити IT-інфраструктуру та виявляти потенційні ризики безпеці.

Інструменти інформаційної безпеки та управління подіями служать основою для організацій, які постійно контролюють свою IT-інфраструктуру, виявляючи потенційні загрози безпеці. А втім, не кожна організація вважає доцільним використовувати технологію штучного інтелекту в дослідженні кіберзлочинів. Проте вкрай важливо визнати, що інструменти кібербезпеки, позбавлені штучного інтелекту, мають власний набір переваг [77]:

1. Економічна ефективність: інструменти, не пов'язані зі штучним інтелектом, часто мають нижчу ціну, ніж аналоги на основі штучного інтелекту, що робить їх економічно вигіднішим вибором для невеликих організацій або тих, хто працює в умовах суворих бюджетних обмежень.

2. Простота використання: інструменти без штучного інтелекту, як правило, простіші в налаштуванні та експлуатації, вимагають менше технічних знань як для налаштування, так і для обслуговування.

3. Налаштування: ці інструменти пропонують вищий ступінь налаштування, дозволяючи організаціям модифікувати правила кореляції та механізми виявлення, які точно відповідають їхнім унікальним умовам безпеки.

4. Гнучкість: інструменти, не пов’язані зі штучним інтелектом, демонструють більший ступінь гнучкості, дозволяючи бездоганну інтеграцію з іншими інструментами та процесами безпеки, сприяючи цілісній безпеці.

Тому, для організацій, які мають обмежений бюджет та ресурси, може бути не доцільним використання штучного інтелекту. Проте для великих організацій з більшими можливостями штучний інтелект може стати у нагоді, так як має такі переваги [78]:

1. Автоматизація: рішення на основі штучного інтелекту можуть автоматизувати багато завдань, зменшуючи робоче навантаження на команди безпеки та дозволяючи їм зосередитися на більш складних завданнях.

2. Розширене виявлення загроз: рішення на основі штучного інтелекту можуть аналізувати великі обсяги даних і виявляти складні загрози, які можуть бути пропущені традиційними інструментами [78].

3. Підвищена точність: рішення на основі штучного інтелекту можуть зменшити помилкові спрацювання та помилкові негативні результати, підвищуючи точність виявлення загроз і зменшуючи втому від повідомлень з систем виявлення вторгнень.

4. Моніторинг у режимі реального часу. Рішення на основі штучного інтелекту можуть забезпечувати моніторинг у режимі реального часу та сповіщення, що дозволяє командам безпеки швидко реагувати на потенційні загрози.

Звичайні інструменти, які не використовують штучний інтелект і рішення на основі штучного інтелекту мають свої власні переваги та недоліки. Найбільш прийнятний вибір для організації буде залежати від її конкретних вимог і фінансових ресурсів. Хоча рішення на основі штучного інтелекту гарантують розширені можливості виявлення загроз і автоматизації, інструменти SIEM без штучного інтелекту можуть виявитися більш економічними та гнучкими. Незважаючи на те, що моделі штучного інтелекту можуть позитивно впливати на безпеку інформаційних систем, важливо визнати, що вони потребують постійного контролю командами безпеки. Також інструменти штучного інтелекту повинні

розглядатись організаціями як засоби, що дозволяють командам безпеки автоматизувати рутинні завдання та покращити ефективність роботи центрів моніторингу, а не як заміну персоналу, адже алгоритми штучного навчання потребують постійного навчання та корегування за потреби.

## 2.4. Дослідження можливостей використання чат-ботів з використанням моделі GPT для аналізу журналів подій

GPT (Generative Pre-trained Transformer) – це алгоритм обробки природної мови, створений американською компанією OpenAI. Головна особливість цього алгоритму полягає в його здатності запам'ятовувати та аналізувати інформацію. Завдяки можливостям обробки природної мови GPT може розуміти, класифікувати та аналізувати журнали подій. Завдяки здатності аналізу журналів подій GPT може виявляти аномалії, надавати розуміння поведінки системи та створювати звіти. Це сприяє швидшому виявленню проблеми, аналізу першопричини та допомагає автоматизувати процес моніторингу [79].

Для дослідження можливостей чат-ботів з використанням GPT було використано вразливе середовище, основним компонентом якого є OWASP Juice Shop, встановлений на сервері Ubuntu в хмарному середовищі Digital Ocean [80]. Детальна схема налаштування середовища вказана на Рис. 2.8.

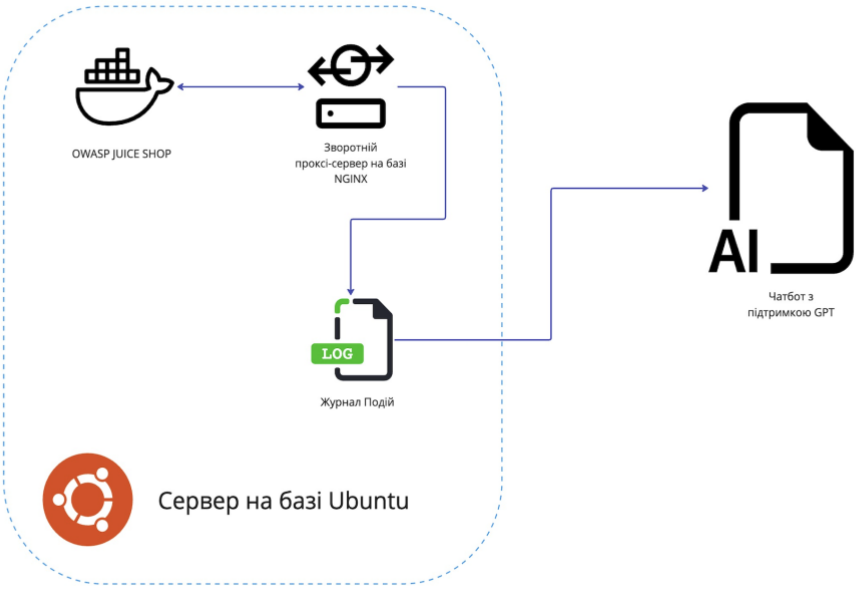



Рис. 2.8. Схема потенційно вразливого середовища

Для підготовки середовища було проведено наступні дії:

1. Встановлено сервер Ubuntu в Digital Ocean, інстальовано контейнер Owasp Juice Shop та налаштовано Nginx як Reverse Proxy для запису всіх запитів до веб-серверу.

2. Для створення журналів подій з потенційними атаками на налаштоване середовище використовували наступні застосунки: Nessus, nmap, BurpSuite та Metaspoit. Автоматизовані засоби пошуку вразливостей використовували для охоплення якомога більшого типу атак.

3. Для дослідження різних типів кібератак за допомогою GPT-3.5 та GPT-4.0 було використано журнали подій, згенеровані після сканувань та експлуатації вразливостей та запропонований варіант реалізації алгоритму GPT компанією OpenAI - ChatGPT.

Під час першого експерименту алгоритмам GPT-3.5 та GPT-4.0 було надано журнали подій, що містять для аналізу атаку обходу каталогу. Атака обходу каталогу (також відома як атака на обхід шляху або перехід на каталог) є типом кібератаки, яка використовує вразливі місця в механізмах обробки файлів веб-додатку [81]. Зловмисник прагне отримати неавторизований доступ до файлової системи програми, дозволяючи читати, змінювати або видаляти конфіденційні файли та дані. Усі запити в даному експерименті сформовані англійською.

Для обох моделей GPT було запропоновано по 200 ідентичних записів подій з журналів подій: по 100 закодованих та 100 нормалізованих записів подій. Кожній з моделей було надано по 10 подій щогодини протягом 10 годин. Приклад записів подій наданих для аналізу:

- *92.253.XX.XX -- [13/Mar/2023:21:33:39 +0000] "GET……../etc/passwd (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"HTTP/1.0" 404 564 "-" "Mozilla/4.0*
- *92.253.xxx.xxx - [13/Mar/2023:21:33:39 +0000] "GET /%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uffoe/%uff0e%uff0e/%uff0e%uff0e/%uff0e%uff0e/etc/passwd HTTP/1.0" 400 166 "-" "-"*

За допомогою алгоритму було проаналізовано та описано записи, виконані зловмисником. Середній час аналізу запиту без кодування з 10 спроб GPT 3.5 - 7.9 секунд, закодованого - 8.3 секунд. У випадку з GPT 4 - 5.8 секунд, закодованого - 6.2 секунд.

Наступна атака, яку було запропоновано ChatGPT для аналізу, – це міжсайтовий скриптинг. Міжсайтовий скриптинг (XSS) – це атака, спрямована на вразливість веб-застосунку, що дозволяє зловмисникам виконувати шкідливі скрипти на веб-сторінках, які відкривають користувачі. Дана вразливість виникає, коли програма випадково включає неперевірені дані (як правило, введені користувачем) на веб-сторінку без відповідної перевірки, захисту або кодування. Як наслідок, шкідливий скрипт зловмисника запускається у веб-браузері жертви, що може призвести до несанкціонованого доступу до даних, зламу облікового запису або інших шкідливих дій [82]. Модель GPT-4.0 та GPT-3.5 були використані для аналізу XSS-атаки. Моделям було надано по 100 ідентичних записів журналів подій: по 10 записів щогодини протягом 10 годин. Приклад запису журналів подій:

- *92.253.xx.xx - - [13/Mar/2023:21:33:53 +0000] "GET /xmd79sr7.asp?<script>document.cookie=%22testmtbo=2804;%22</script> HTTP/1.0" 404 564 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"*

ChatGPT проаналізував атаку та способи використання XSS. Було встановлено, що зловмисник мав на меті маніпулювати програмою за допомогою параметрів JavaScript і URL-адреси. У журналах подій було виявлено запити GET, спрямовані до файлу "xmd79sr7.asp", які супроводжувались параметрами URL-адреси, що містили шкідливий JavaScript вміст. Моделі визначили наступний запис, як шкідливий: "<script>document.cookie=%22testmtbo=2804;%22</script>". Також, ChatGPT було зроблено висновок, що зловмисник хотів, щоб програмне забезпечення представило та згодом запустило цей код у браузері користувача. Подальший аналіз ChatGPT відповіді сервера виявив, що відповідь містила код статусу 404, що означає “Не знайдено”. Це означає: або веб-сервер не обробив запит, або наявні налаштування сервера пом’якшують такі загрози. Середній час

аналізу з допомогою GPT-4.0 з 10 спроб становив 12.3 секунд та GPT-3.5 – 17.5 секунд.

Також в рамках цього дослідження проаналізовано можливість виявлення атаки типу сканування на вразливості. Для цього було використано множину записів з журналів подій – по 10 однакових записів кожній моделі, щогодини по 10, які надали для аналізу моделям GPT, зокрема містили наступні записи, що свідчать про сканування на вразливості:

- *92.253.xx.xx- - [13/Mar/2023:21:33:52 +0000] "POST /spipe?Source=nessus HTTP/1.0" 404 564 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"*
- *92.253.xx.xx - - [13/Mar/2023:21:33:52 +0000] "POST /cgi-bin/mainfunction.cgi HTTP/1.0" 404 564 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"*
- *92.253.xx.xx - - [13/Mar/2023:21:34:07 +0000] "POST /flex2gateway/http HTTP/1.0" 404·564 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"*

У дослідженні використовувались моделі GPT-4.0 та GPT-3.5. ChatGPT оцінив, що надані журнали подій свідчать про спроби зловмисників визначити слабкі місця веб-сервера за допомогою різних запитів. Середній час аналізу журналів подій становив 12.9 секунд за допомогою GPT-4.0 та 18.3 секунд для GPT 3.5, при цьому кожен запит розглядали окремо. Варто зауважити, що запити оцінювалися та відповіді надавалися українською мовою.

Для подальшої оцінки та порівняння моделей були представлені записи журналу подій, отримані сервером під час спроби зловмисником використати вразливість Log4j. Зокрема, журнали подій містили наступні записи:

- *92.253.xx.xx [13/Mar/2023:21:39:23 +0000] "GET /wp-login.php HTTP/1.0" 404 162 "${jndi:ldap://log4shell-generic 8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w.nessus.org/nessus}""${jndi:ldap://log4shell-generic-8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w.nessus.org/nessus}"*

- *92.253.xx.xx - - [13/Mar/2023:21:39:23 +0000] "GET /wp-login.php HTTP/1.0" 404 162"${jndi:ldap://log4shell-generic 8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w. nessus.org/nessus}""${jndi:ldap://log4shell-generic 8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w.nessus.org/nessus}"*
- *92.253.xx.xx - - [13/Mar/2023:21:39:28 +0000] "GET /wwwadmin.cgi HTTP/1.0" 404 162 "${jndi:ldap://log4shell-generic-8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w.nessus.org/nessus}""${jndi:ldap://log4shell- generic8Vno2Ky4QW5hhAz16ZUW${lower:ten}.w.nessus.org/nessus}"*

Log4Shell (CVE-2021-44228) є критичною вразливістю, виявленою у бібліотеці Apache Log4j, що дозволяє зловмиснику віддалено виконати шкідливий код [83]. Обидві моделі виявили спробу експлуатації CVE-2021-44228. Середній час аналізу з 10 спроб проведених за 10 годин для GPT 4.0 – 15.3 секунд, та GPT-3.5 – 18.1 секунд.

Отож, у результаті цих експериментів визначено, що з усіх наданих записів журналів подій моделі GPT-3.5 та GPT-4.0 опрацювали та успішно виявили усі типи атак, що підтверджено проведеними експериментами з використанням масивів журналів подій. Для порівняння моделей результати експерименту зазначено у Таблиці 2.2. Для аналізу даних було визначено різницю швидкості аналізу у відсотковому відношенні та сформовано діаграму порівняння моделей між собою, зображену на Рис.2.8.

Таблиця 2.2.

Порівняльна характеристика GPT-3.5 та GPT-4.0 для аналізу подій

| Атака | Середній час аналізу з 10 спроб (GPT-3.5) | Середній час аналізу з 10 спроб (GPT-4) | Різниця швидкості аналізу % |
|---|---|---|---|
| Атака обходу каталогу нормалізований | 6,2 | 5,8 | 6.45% |

Продовження таблиці 2.2.

| | | | |
|---|---|---|---|
| XSS атака | 6,4 | 5,7 | 10.94% |
| Атака обходу каталогу (закодований запит) | 8,3 | 7,9 | 4.82%. |
| Атака сканування на вразливості | 17,5 | 12,3 | 29.71% |
| Експлуатація CVE-2021-44228 | 18,1 | 15,3 | 15.47% |

GPT-4.0 загалом демонструє підвищену ефективність обробки та виявлення різних типів кібератак порівняно з GPT-3.5. Серед усіх протестованих типів атак GPT-4.0 стабільно демонструє швидший час відповіді. Зокрема, для атаки зі скануванням вразливостей і використання CVE-2021-44228 GPT-4.0 до 29,71% і 15,47% швидше аналізує журнали подій, ніж GPT-3.5 відповідно. Ці дані свідчать про те, що GPT-4.0 може бути більш ефективним при аналізі складніших типів атак. Для веб-атак, таких як обхід каталогу (як нормалізований, так і закодований) і атаки XSS, GPT-4.0 швидше до 10,94%.

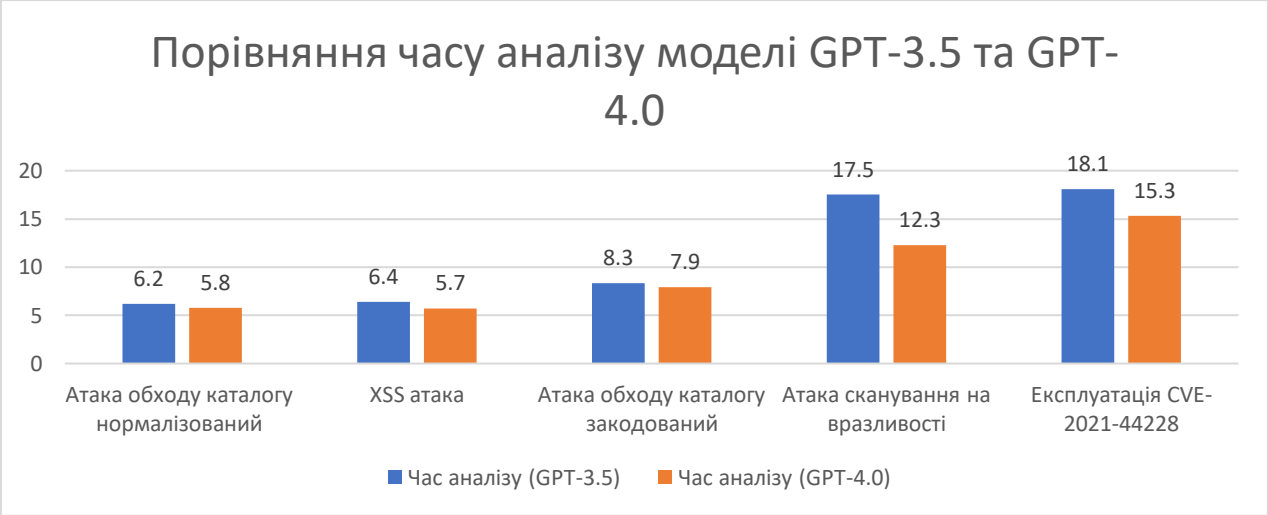



Рис. 2.9. Порівняння часу аналізу моделі GPT-3.5 та GPT-4.0

Проведені експерименти встановили, що GPT-4.0 не лише точно визначає тип кібератаки, але забезпечує загалом швидше їх виявлення, що може мати вирішальне значення під час реальних загроз безпеці інформаційній системі, де час відповіді потрібно мінімізувати.

## 2.5. Концепція моделі дослідження кіберзлочинів

Після проведених досліджень, що були зроблені у попередніх розділах, сформовано концепцію моделі системи дослідження кіберзлочинів. Ця концепція представлена на Рис. 2.9 та складається з наступних компонентів:

1. Система дослідження загроз – це зовнішня система, що відповідає за збір індикаторів компрометації систем (ІКС), зокрема шкідливих IP, хеш-сум, доменних імен, імейлів. Система дослідження загроз надає інформацію компоненту аналізу подій про потенційні індикатори компрометації систем.
2. Система управління вразливостями – це зовнішня система, яка відповідає за консолідацію інформації про вразливості, визначені для інформаційної системи та надсилання цієї інформації до компоненту аналізу подій.
3. Компоненти статичного тестування безпеки додатків (SAST), динамічного тестування безпеки додатків (DAST), аналізу складників програмного забезпечення (SCA) є зовнішніми системами, що використовують розробники програмного забезпечення для аналізу вразливостей програмного забезпечення. Компонент аналізу подій отримує інформацію з цих систем.
4. Модель GPT – стороння система, з’єднання з якою забезпечено за допомогою API інтерфейсу. Компонент аналізу подій надсилає консолідовану та масковану інформацію про потенційний кіберзлочин цій системі.
5. Компонент виявлення аномалій – внутрішній алгоритм системи дослідження кіберзлочинів, що відповідає за дослідження нормальної активності користувачів, аналізуючи події інформаційної системи, та виявлення незвичних дій.
6. Компонент аналізу подій – внутрішній алгоритм системи дослідження кіберзлочинів, що відповідає за консолідацію інформації з різних джерел даних та аналіз подій за допомогою сторонніх сервісів з підтримкою GPT.

7. Компонент маскування даних – внутрішній алгоритм, що відповідає за маскування даних системи дослідження кіберзлочинів.

8. Компонент представлення – інтерфейс роботи з системою, який використовує користувач.

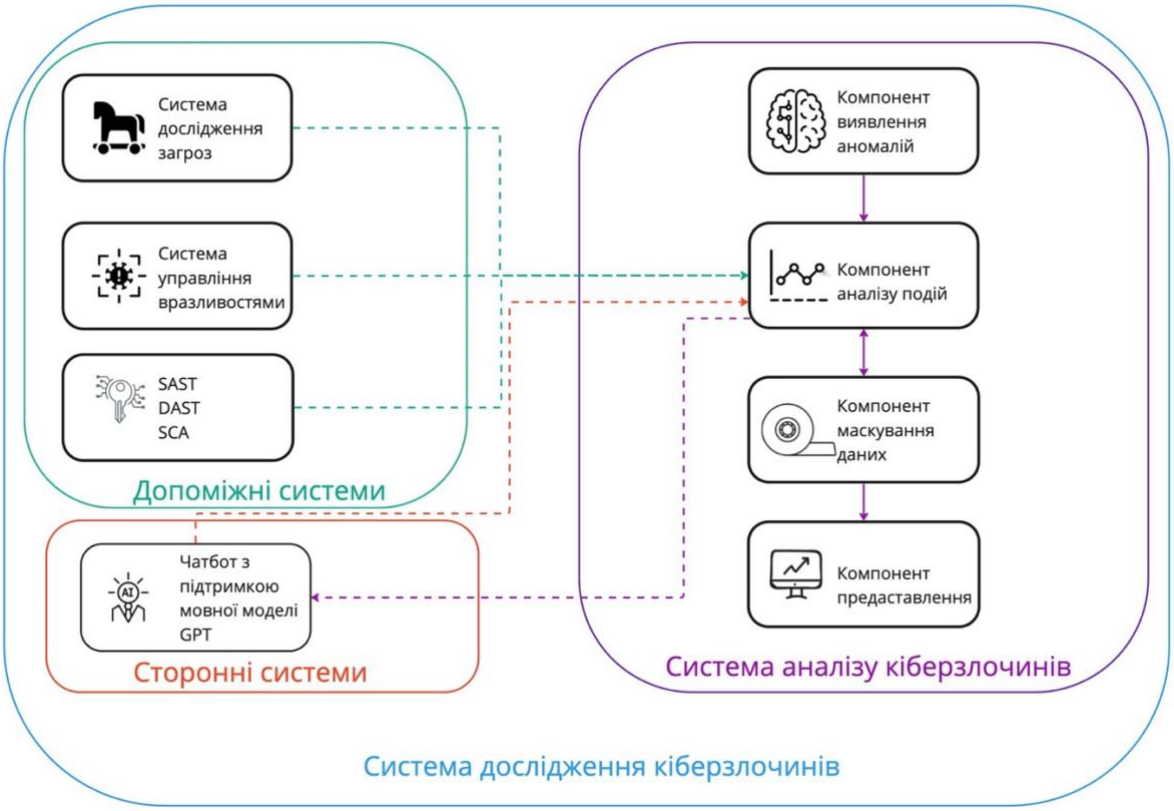



Рис. 2.10. Схематичне зображення компонентів системи дослідження кіберзлочинів

Отже, для побудови системи дослідження кіберзлочинів було визначено усі необхідні компоненти. Детальний опис компонентів та їх цілей надано у третьому розділі наукової роботи.

## 2.6. Використання технології Blockchain для дослідження кіберзлочинів

Для збору журналів подій для системи дослідження кіберзлочинів запропоновано використовувати систему приманок на основі технології Blockchain. Система використовує динамічні атрибути технології Blockchain для зміни сервісів, що використовуються вузлами. Дії зловмисника записують у журнал подій, згенерованих системою [84]. Схематично система позначена на Рис. 2.10., а детальний опис принципу роботи системи та генерації подій подано нижче.

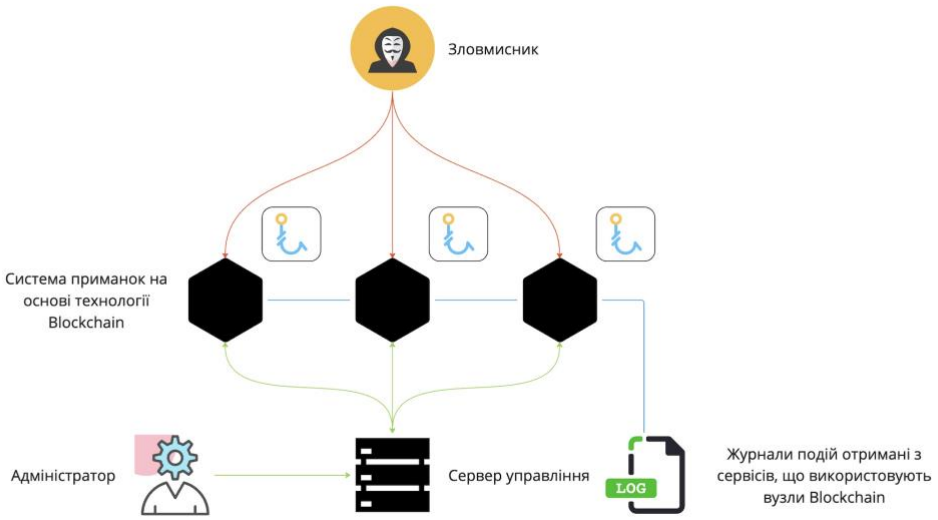



Рис. 2.11. Схема системи приманок на основі технології Blockchain

Система виконує створення, відправлення, отримання, очікування, відкриття, закриття, записування, відновлення та компрометацію. Позначення, які використовуються для системи, описані в таблиці 2.3. Дії системи та їх параметри зведені в таблицю 2.4.

Таблиця 2.3.

Позначення, які використовуються для системи приманок на основі технології Blockchain

| Назва | Позначення |
|---|---|
| Стан системи | $\{st_n, st_c\}$ |
| Дані | $\{d_1, d_2, \ldots, d_n\}$ |
| Шкідливі дані | $\{md_1, md_2, \ldots, md_n\}$ |
| Канали передачі даних | $\{c_1, c_2, \ldots, c_n\}$ |
| Активи | $\{a_1, a_2, \ldots, a_n\}$ |
| Сервіси | $\{s_1, s_2, \ldots, s_n\}$ |
| Запис журналу подій | $\{log_1, log_2, \ldots, log_n\}$ |

Таблиця 2.4.

Опис дій системи

| Функція | Опис |
|---|---|
| generate(d) | актив створює дані |
| send(d,ch) | актив передає дані через канал передачі даних |
| receive(d,ch) | актив приймає дані через канал передачі даних |
| deny(d,ch) | актив відхиляє/приймає дані через канал передачі даних |
| compromise() | актив піддається компрометації |
| record(log) | актив створює запис у журнали подій |
| await(відповідь) | актив чекає на відповідь після відправки |

Метод збору журналів подій з приманок на основі Blockchain описано нижче:

1. Множина сервісів $\{s_1, s_2, \ldots, s_n\}$ активу отримує запити від зловмисника. Функції generate(d) і send(d) представляють маніпуляції з вихідними даними сервісів, а функція приймання (d) представляє їх отримання ззовні. Після отримання запиту актив може працювати в звичайному або скомпрометованому режимі. Звичайний режим означає, що зловмисник не вплинув на актив і актив підтримує нормальну роботу. Однак скомпрометований режим вказує на те, що зловмисник отримав несанкціонований доступ до активу. Стани $\{st_n, st_c\}$ поділяються на дві категорії: для нормального режиму - $st_n$ та для скомпрометованого режиму - $st_c$. Варто зазначити, що дані, які передають

активами, гарантують нормальну роботу системи, вони відіграють важливу роль в аналізі безпеки [85]. Якщо актив отримує шкідливі дані від зловмисника, він стає скомпрометованим:

$\{a_1, a_2, \ldots, a_n\}$, receive$(md) \longrightarrow$ compromise $\{a_1, a_2, \ldots, a_n\} \wedge st_n, \{a_1, a_2, \ldots, a_n\}$

2. Передбачається, що якщо звичайний актив отримує шкідливі дані, він увійде в режим компрометації, далі компрометуючи себе. Щоб скомпрометований актив не міг перехопити дані між іншими активами, канали передачі даних $\{c_1, c_2, \ldots, c_n\}$ повинні бути захищені [86].

3. Усі стани в активах ілюструють загальний стан системи, тобто безперервна авторизована поведінка кожного активу забезпечує нормальне функціонування системи. Будь-які активи спілкуються одне з одним, надсилаючи та отримуючи дані через канали зв'язку. Це вказує на те, що спільний доступ до одного каналу дозволяє як з'єднуватися, так і передавати спільні дані. Обмін даними може здійснюватися тільки при підключенні через один канал зв'язку [87]. Отже, ми визначаємо таке твердження: актив генерує дані та надсилає їх активу через канал передачі інформації.

$$a_1 \longrightarrow generate(d_1) \wedge send(d_1)$$

4. Отримавши дані від активу $a_1$, актив $a_2$ може вирішити, прийняти чи відхилити ці дані. Коли актив $a_2$ отримує та приймає шкідливі дані, він стає скомпрометованим [88].

$$a_1, send(c, d_1) \rightarrow a_2, receive(c, d_1)$$

$$a_1, send(c, d_1) \rightarrow a_2, deny(ch, d_1)$$

Якщо: $d_1 = md$

Тоді: $a_1, compromise$

Отже: $sc(a_1)$

5. Звичайний актив є легальною частиною системи та забезпечує нормальне функціонування її сервісів для користувачів. Система адаптується та зосереджується на передачі даних для подальшого аналізу атак та записує всі дані про атаку у журнал подій [89].

$$a_2 \longrightarrow \text{record(log)} \wedge \text{await(response)}$$

Журнали подій, згенеровані системою приманок на основі технології Blockchain, використовують у цьому дослідженні для аналізу подій системою дослідження кіберзлочинів. Процес створення тестових даних описаний у Розділі 4 наукової роботи.

**Висновки до розділу 2**

У другому розділі проаналізовано можливості використання різних типів систем дослідження подій інформаційної безпеки, проаналізовано використання алгоритмів машинного навчання для дослідження аномалій у журналах подій та використання алгоритмів GPT для аналізу кібератак. Дані дослідження були проведені для обґрунтування припущення, що моделі штучного інтелекту можуть бути використані для виявлення та аналізу кіберзлочинів, що підтверджено описаними у цьому розділі експериментами. Окрім цього, проведене дослідження дозволяє виділити наступне:

1. З огляду на виявлені проблеми сучасних систем дослідження подій, такі як управління даними, надлишок сповіщень, складності інтеграції, питання конфіденційності, відповідності та динамічно змінюваних загроз, а також обмежену здатність виявлення загроз нульового дня через залежність від статичних правил, існує потреба у розвитку нового покоління систем виявлення вторгнень. Ці системи мають бути оснащені можливостями машинного навчання для гнучкого аналізу та адаптації до кіберзагроз, що еволюціонують, здатні ефективно управляти повідомленнями та інтегруватися з іншими безпековими рішеннями, а також забезпечити більшу прозорість та конфіденційність у роботі з даними.

2. Проведений ґрунтовний аналіз переваг та недоліків використання штучного інтелекту дозволив визначити, що модель дослідження кіберзлочинів на основі штучного інтелекту може бути використана під час управління інцидентами інформаційної безпеки, зокрема при виявленні та аналізі подій. Визначено, що основні принципи використання штучного інтелекту для таких процесів полягають у здатності до глибинного аналізу даних, швидкості виявлення зловмисних дій та ефективності у класифікації й оцінці інцидентів. Встановлено, що моделі штучного інтелекту можуть автоматизувати виявлення складних зразків поведінки та

аномалій, які можуть вказувати на інформаційні загрози або вторгнення, забезпечуючи у такий спосіб більш оперативне управління інцидентами.

3. Порівняльний аналіз алгоритмів машинного навчання визначив, що різні моделі машинного навчання можуть бути використані для виявлення аномалій інформаційної безпеки за належного тренування. Ізоляційний ліс виділявся своєю здатністю розрізняти аномалії та низьким рівнем хибних спрацювань зокрема з найвищими показниками міри роздільності (AUC) – 0,65, адже що вищий AUC, то краще модель розрізняє позитивні та негативні класи. Ізоляційний ліс також продемонстрував найвищий показник влучності (052), що є відношенням точно позитивних до суми хибних та точно позитивних результатів Також ізоляційний ліс виділився найвищим показником оцінки F1 (0,56), що є середнім гармонійним значенням влучності та відкликання. Тоді як випадковий ліс вирізнявся відкликанням, але мав проблеми з опрацюванням аномалій, наданих наборів даних. Модель глибокого навчання забезпечила збалансовану продуктивність, але не найкращу точність та найвищу кількість хибних спрацювань. Зважаючи на дані проведених досліджень та аналіз роботи моделей, було прийнято рішення використовувати модель ізоляційного лісу як елементу моделі дослідження кіберзлочинів.

4. Дослідження використання моделей GPT-4.0 дало можливість встановити, що ці моделі не лише точно визначають тип кіберзлочину, але гарантують швидке рішення для виявлення кібератак, що може мати вирішальне значення в реальних сценаріях, де час відповіді необхідно мінімізувати. Зокрема, GPT-4.0 загалом демонструє підвищену ефективність обробки та виявлення різних типів кібератак порівняно з GPT-3.5. Серед усіх протестованих типів атак GPT-4.0 стабільно демонструє швидший час відповіді. Зокрема, для атаки зі скануванням вразливостей і використання CVE-2021-44228 GPT-4.0 до 29,71% і 15,47% швидше аналізує журнали подій, ніж GPT-3.5 відповідно. Ці дані свідчать про те, що GPT-4.0 може бути більш ефективним при аналізі складніших типів атак. Для веб-атак, таких як обхід каталогу (як нормалізований, так і закодований) і атаки XSS, GPT-4.0 швидше до 10,94%. Також було визначено обмеження публічних систем з

використанням GPT, не усі системи можуть гарантувати приватність даних, які їм передають, що вказується у політиках застосунків, тому є необхідність встановити додаткові контролі безпеки перед передачею даних сторонній системі.

# РОЗДІЛ 3. РОЗРОБКА МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ ДЛЯ СКЛАДОВИХ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ

## 3.1. Визначення вимог міжнародних стандартів до системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем

У цьому розділі наведено огляд ключових компонентів, які утворюють комплексну систему розслідування кіберзлочинів, висвітлюючи технологічні та процедурні елементи, необхідні для ефективної реалізації системи.

Одним із завдань, що забезпечує ефективність та інноваційний підхід у дослідженні кіберзлочинів, є покращення процесу управління інцидентами інформаційної безпеки. Для узгодження цього процесу з кращими світовими практиками та впровадження системи державними та приватними організаціями модель системи дослідження кіберзлочинів повинна відповідати міжнародним стандартам. Популярними стандартами, які можуть бути застосовні для систем дослідження кіберзлочинів, можуть бути:

1. Спеціальна публікація NIST 800-53 “Безпека і засоби контролю конфіденційності для інформаційних систем і організації” з національного інституту стандарти та технології (NIST) наразі діє його 5-та редакція (rev5) від вересня 2020 року. Спочатку даний стандарт розроблений для захисту даних федерального уряду США, але швидко здобув популярність серед приватної промисловості і зараз розглядається, як один з найпопулярніших та респектабельних інформацій системи безпеки у світі. Це сталось частково через значний аутсорсинг у приватні компаній, які мають справу з урядовими організаціями США [90].

2. PCI DSS стандарт розроблений для захисту інформації про кредитні та дебетові картки. Стандарти безпеки даних (DSS) індустрії платіжних карток (PCI) описують основні принципи і вимоги інформаційної безпеки, яких необхідно дотримуватися, включаючи запобіжні заходи щодо програмного забезпечення, яке обробляє дані кредитних карток [91].

3. Міжнародний стандарт ISO/IEC 27001:2022 визначає вимоги до створення, впровадження, підтримки та постійного розвитку системи менеджменту інформаційної безпеки. Даний стандарт також містить детальний перелік контролів інформаційної безпеки, які організації можуть адаптувати відповідно до проведеної попередньої оцінки ризиків [92].

Серед вказаних стандартів ISO 27001 виділяється своєю універсальністю і комплексним підходом до захисту інформації, адже ISO 27001 не обмежений географією чи галуззю та забезпечує систематичну структуру, яка включає процеси управління ризиками, засоби контролю безпеки та заходи відповідності, застосовні до всіх типів інформаційних активів. На відміну від стандарту NIST, який є набором інструкцій, ніж стандартом сертифікації, або PCI DSS, який вузько орієнтований на дані платіжних карток, ISO 27001 вимагає від організацій впровадження широкого набору заходів безпеки інформації та постійного вдосконалення системи менеджменту інформаційної безпеки. Це гарантує цілісну безпеку, яка охоплює не лише цифрову кібербезпеку, а також процедурні аспекти інформаційної безпеки. Крім того, відповідність стандарту ISO 27001 демонструє зацікавленим сторонам відповідність кращим практикам інформаційної безпеки, яка визнана в усьому світі, на відміну від інших стандартів, які можуть мати більш регіональну спрямованість або обмежену сферу дії. Тому у цій науковій роботі система дослідження кіберзлочинів визначає контролі стандарту ISO 27001:2022, на які вона може мати вплив. Це дає можливість узгодити процеси, запропоновані системою, з найкращими практиками управління інформаційною безпекою.

З огляду на все вищезгадане, під час розробки системи дослідження кіберзлочинів було визначено наступні контролі інформаційної безпеки, які потрібно забезпечити під час її розробки:

1. Система дослідження кіберзлочинів повинна забезпечити автоматичний аналіз даних з різних джерел для визначення характеристик і масштабу інцидентів безпеки (ISO/IEC 27001:2022 A.5.25).

2. Система повинна забезпечити автоматизацію реагування на інциденти (ISO 27001:2022 A.5.26).

3. Моделі штучного інтелекту повинні забезпечити навчання після інциденту (ISO 27001:2022 A.5.27). За допомогою ШІ система повинна забезпечити поглиблений аналіз даних про інциденти після їх усунення, щоб сформувати стратегії запобігання та реагування.

4. Система повинна виконувати звітування про інциденти та документування (ISO/IEC 27001:2022 A.5.28), щоб допомогти у створенні вичерпних і узгоджених звітів про інциденти, забезпечуючи детальну та точну документацію для відповідності та перегляду.

### 3.2. Розробка моделі системи дослідження загроз

Одним із компонентів системи дослідження кіберзлочинів є система дослідження загроз. Основним завданням цього компоненту у системі дослідження кіберзлочинів є збір індикаторів компрометації систем з джерел загроз для виявлення потенційних або ж підтверджених кібератак та кіберзлочинів.

Принцип роботи системи дослідження загроз можна описати таким чином. Позначимо алгоритм збору даних компрометації систем як функцію A, яка працює на системі дослідження загроз, у якій SL – це список джерел (Source List, скорочено SL), а TI – це система дослідження загроз (Threat Intelligence, скорочено TI). ∑: вказує на підсумковий процес ітерації кожного джерела в списку. n: загальна кількість джерел. $\text{Source}_i$ представляє джерело зі списку. $\text{Verify}(\text{Source}_i)$ представляє етап перевірки для кожного джерела (перевірка формату даних і сумісності API). GetCred етап отримання облікових даних для кожного джерела. TestCon – перевірка з'єднання з джерелом за допомогою отриманих облікових даних. Integrate(): ця функція представляє інтеграцію джерела в системі дослідження загроз, якщо джерело перевірено, облікові дані дійсні та перевірка підключення пройшла успішно. Отож, кінцева формула запропонованого алгоритму виглядає так:

$$A(\text{TI},\text{SL}) = \sum_{i=1}^{n} \text{Integrate}\big(\text{Verify}(\text{Source}_i),\text{GetCred}(\text{Source}_i),\text{TestCon}(\text{Source}_i)\big), \quad (3.1)$$

Цей алгоритм гарантує безпечну інтеграцію нових джерел розвідки про загрози в систему дослідження загроз. Він ретельно перевіряє джерела, обробляє

облікові дані, перевіряє з'єднання, а потім інтегрує та планує синхронізацію, водночас обробляючи будь-які помилки, які виникають під час процесу. Цей систематичний підхід гарантує, що до платформи додаються лише надійні та сумісні джерела, розширюючи її можливості щодо моніторингу та реагування на кіберзагрози.

Для перевірки наявності індикатора компрометації у системі дослідження загроз спостережувані дані порівнюють з наявним масивом індикаторів компрометації систем (IOC). Якщо значення MatchScore ObservedData проти IOCs більше або дорівнює пороговому значенню(Threshold): принаймні 1, то наявність індикатора компрометації у системі дослідження загроз підтверджена. Формула перевірки наявності індикаторів компрометації виглядає таким чином:

$$\text{VerifiedIOC} = \text{TrueifMatchScore}(\text{IOCs}, \text{ObservedData}) \geq \text{Threshold}, \text{False}, \quad (3.2)$$

Цей підхід було реалізовано на системі з відкритим кодом для дослідження загроз MISP. Запропонована структура системи дослідження загроз зображена на Рис. 3.1.

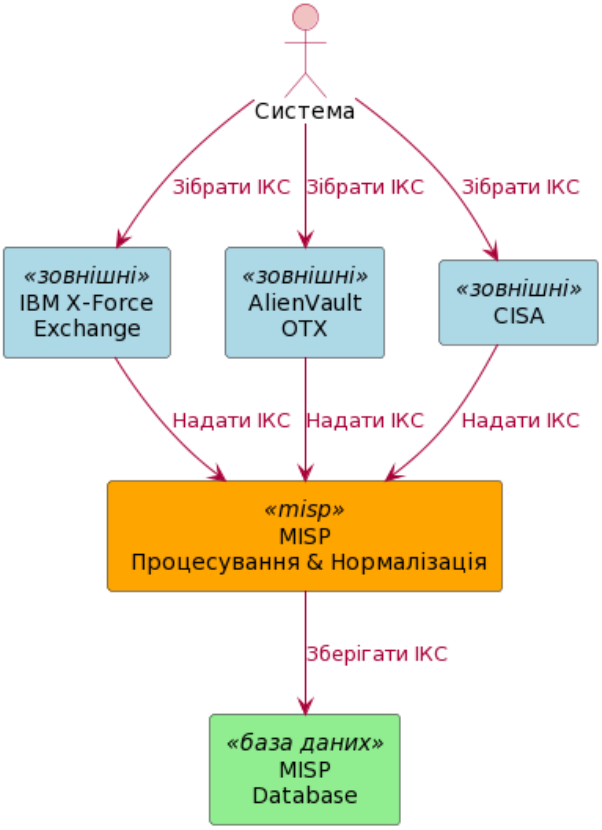



Рис. 3.1. Схематичне зображення системи дослідження загроз

На цьому рисунку зображено взаємодію джерел розвідки про загрози та системи дослідження кіберзлочинів. Основою системи дослідження загроз було визначено систему MISP з відкритим кодом. Зовнішні джерела загроз, які надсилають індикатори компрометації систем, включають IBM X-Force Exchange, AlienVault OTX та CISA автоматичне розповсюдження індикаторів компрометації систем.

Отримання даних про загрози починається з циклічного проходження по масиву джерел загроз. Кожне джерело містить такі важливі деталі, як ім’я, URL-адресу, ключ доступу, формат даних, частоту синхронізації та правила фільтрації. Для кожного джерела алгоритм перевіряє:

1. Чи формат даних джерела є одним із прийнятних форматів? Наприклад: STIX, TAXII, CSV або JSON.

2. Чи API джерела сумісний із системою? Тобто чи може він ефективно спілкуватися з платформою.

Потім алгоритм намагається отримати необхідні облікові дані (наприклад, ключі API) для кожного джерела. Він перевіряє, чи ці облікові дані є дійсними та чи можуть вони успішно встановити з’єднання з URL-адресою джерела.

Якщо джерело проходить перевірку та облікові дані дійсні, алгоритм готується до інтеграції джерела в системі MISP. Це передбачає компіляцію таких деталей, як ім’я джерела, URL-адреса, облікові дані та формат даних. Потім він додає ці деталі до системи дослідження загроз, ефективно інтегруючи нове джерело. Після успішного додавання джерела алгоритм встановлює розклад для синхронізації з джерелом. Ця синхронізація відбувається з частотою, зазначеною в деталях джерела, містить спеціальні правила фільтрації для керування вхідними даними.

### 3.3. Компонент управління вразливостями

Для дослідження вразливостей, що можуть бути використані зловмисником для компрометації інформаційної системи та аналізу впливу невиправленої вразливості на виникнення події інформаційної безпеки, було розроблено систему управління вразливостями. Дана система описана набором наступних функцій.

1. Функція виявлення вразливостей:

$$VD(S) = \sum_{i=1}^{n} v_i(S), \tag{3.3}$$

У якій VD(S) – це функція виявлення вразливостей системи, (S) представляє окрему вразливість, виявлену в системі, а n – загальна кількість вразливостей, виявлених в інформаційній системі.

2. Функція оцінки вразливостей:

$$VS(v_i) = (\text{CVSS v3.0}), \tag{3.4}$$

Для оцінки вразливості пропонують використовувати загальну систему оцінки вразливостей (CVSS v3.0), яка є стандартизованою системою оцінки серйозності вразливостей системи безпеки. У даному випадку VS(vi) є оцінкою вразливості, що базується на стандартизованому значенні CVSS v 3.0 визначеного системою сканування. Проте CVSS 3.0 не враховує індивідуальні характеристики інформаційної системи, які можуть впливати на серйозність вразливості та першочерговість виправлення, зокрема фазу розробки системи, наявність заходів безпеки та критичність даних. Тому для врахування даних факторів запропоновано ввести ваговий коефіцієнт, що може бути застосований організаціями за потреби для покращення оцінки дослідження кіберзлочинів, спричинених невиправленою вразливістю. На основі експертних суджень було визначено оцінку для кожного з критеріїв вагового коефіцієнту. Зібрано результати опитувань 20 експертів інформаційної безпеки та запропоновано оцінки у Таблицях 3.1 – 3.3. Цей ваговий коефіцієнт не є обов'язковим, проте дає можливість аналітикам інформаційної безпеки враховувати індивідуальні характеристики системи, яку досліджують. З урахуванням вагового коефіцієнту загальна оцінка вразливості системи описується такою формулою:

$$VA(S) = \sum_{i=1}^{n} VS(v_i) \times W(\mathrm{P}, \mathrm{SC}, \mathrm{IC}), \tag{3.5}$$

Ця формула передбачає, що $VS(v_i)$ – це CVSS 3.0, а ваговий коефіцієнт, що базується на наступних критеріях: фаза розробки системи (P), наявність заходів безпеки (SC) – загальний коефіцієнт, що обчислюється шляхом множення

коефіцієнтів кожного з заходів безпеки та класифікація інформації (ІС), що обробляються системою. Обґрунтування даних критеріїв та відповідні коефіцієнти запропоновані у Таблицях 3.1-3.3. Ваговий коефіцієнт визначає вплив даних факторів на загальну оцінку вразливостей системи, проте варто зазначити, що дані оцінки мають бути скориговані відповідно до певної толерантності до ризику та політики організації. Представити формулу ваги, враховуючи вищезазначені фактори, можна як середнє значення цих трьох факторів:

$$W = \frac{P + SC + IC}{3}, \tag{3.6}$$

Таблиця 3.1.

Критерій: Фаза розробки інформаційної системи (Р)

| Фаза розробки | Коефіцієнт | Обґрунтування |
|---|---|---|
| В активному використанні | 1.0 | Інформаційні системи в активному використанні зазвичай мають найвищий пріоритет, оскільки вразливі місця в цих системах можуть безпосередньо вплинути на організацію. Порушення або збій тут зазвичай має значний вплив на інформаційну систему та організацію. |
| У розробці | 0.75 | Система у розробці менш критична, ніж в активному використанні, але вона може бути копією системи у використанні або містити конфіденційну інформацію та інтелектуальну власність організації. |

Продовження таблиці 3.1.

| | | |
|---|---|---|
| Підтвердження концепції (POC) | 0.25 | Підтвердження концепції (POC) має найменший вплив, оскільки її часто ізольовано та використовують для експериментальних цілей. Ризик і вплив вразливостей тут зазвичай нижчі порівняно з іншими середовищами. |

Таблиця 3.2.

Критерій: заходи безпеки (SC)

| Наявність заходів безпеки (SC) | Коефіцієнт | Обґрунтування |
|---|---|---|
| Брандмауери та сегментація мережі | 0,8 (впроваджені заходи), 0,9 (частково впроваджені), 1,0 (заходи відсутні) | Брандмауери та сегментація мережі значно зменшують ризик експлуатації вразливостей. |
| Системи виявлення та запобігання вторгненням (IDS та IPS) | 0,7 (впроваджені заходи), 0,85 (частково впроваджені), 1,0 (заходи відсутні) | Ефективні IDS та IPS можуть виявляти та потенційно запобігати використанню вразливостей. |
| Захист кінцевої точки (антивірус, EDR тощо) | 0,75 (впроваджені заходи), 0,9 (частково впроваджені), 1,0 (заходи відсутні) | Захист кінцевих точок може знизити ризик використання вразливості зловмисником за наявності поведінкових сигнатур шкідливого ПЗ або експлойтів. |

Продовження таблиці 3.2.

| | | |
|---|---|---|
| Контроль доступу та автентифікація | 0,8 (впроваджені заходи), 0,9 (частково впроваджені), 1,0 (заходи відсутні) | Засоби контролю доступу та механізми автентифікації обмежують можливість несанкціонованого доступу. |
| Шифрування даних (у стані спокою та під час передачі) | 0,85 (впроваджені заходи), 0,95 (частково впроваджені), 1,0 (заходи відсутні) | Комплексне шифрування захищає дані, навіть якщо інші рівні безпеки порушено. |
| Керування виправленнями | 0,8 (впроваджені заходи), 0,9 (частково впроваджені), 1,0 (заходи відсутні) | Вчасне виявлення та реагування на виправлення унеможливить їх експлуатацію. |
| Управління інформаційною безпекою та управління подіями безпеки (SIEM) | 0,75 (впроваджені заходи), 0,9 (частково впроваджені), 1,0 (заходи відсутні) | Використання базових можливостей SIEM з аналізом у реальному часі дає можливість виявити спроби експлуатації вразливостей та зреагувати на них. |

Таблиця 3.3.

Критерій: класифікація інформації (IC)

| Класифікація інформації | Коефіцієнт | Обґрунтування |
|---|---|---|
| Конфіденційна інформація | 1.0 | Системи, які обробляють конфіденційні дані, оскільки порушення може призвести до значної юридичної, фінансової та репутаційної шкоди організації. |
| Службова інформація | 0.75 | Системи, які обробляють службові дані вважаються ризикованими адже містять стратегічно важливу інформацію про організацію та її інфраструктуру. |
| Публічна інформація | 0.5 | Системи, що працюють із публічною інформацією, вважають менш ризикованими, оскільки дані вже є загальнодоступними, а порушення не призведе до розкриття конфіденційної інформації. Проте маніпуляція з даними такого типу може призвести до поширення неправдивих відомостей, особливо коли таку інформацію використовують у сфері оперативного суспільного інформування або новин. |

Ці формули можуть бути впроваджені в систему дослідження кіберзлочинів, а саме в процес оцінки вразливостей в інформаційних системах. Процес передбачає виявлення окремих вразливостей, оцінювання їх за різними критеріями, а потім узагальнення цих балів для отримання загальної оцінки.

Зважаючи на результати досліджень, проведених у попередніх розділах, було розроблено систему управління вразливостями, що використовує наступні рішення з відкритим кодом: рішення для сканування типу DAST – OWASP ZAP, SCA – OWASP Dependency Check, OpenVAS: інструментів забезпечує комплексний підхід до управління вразливістю, охоплюючи широкий спектр оцінок безпеки від інфраструктури до рівня коду. Схематично систему управління вразливостями представлено на Рис.3.2.

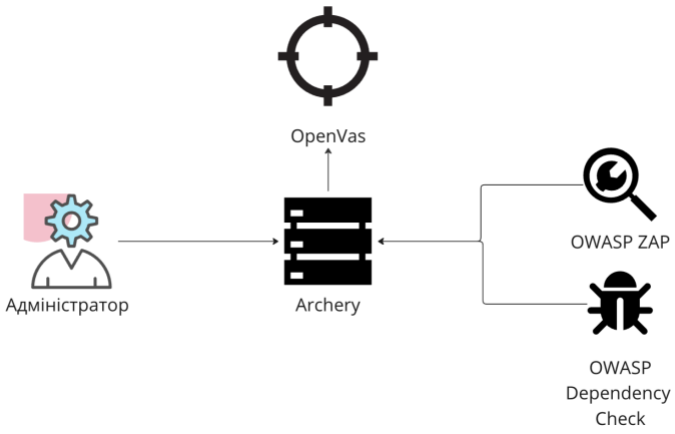



Рис. 3.2. Схематичне зображення системи управління вразливостями

Інтеграція цих інструментів у компонент управління вразливостями дає можливість забезпечити ретельне виявлення вразливостей та керування ними на різних рівнях інформаційних систем.

OpenVAS пропонує широкі можливості сканування інфраструктури та, як інструмент із відкритим вихідним кодом, є економічно ефективним і отримує переваги від оновлень, керованих спільнотою. OWASP Dependency-Check спеціалізується на виявленні вразливостей у залежностях проекту. Він може бути інтегрований у конвеєри CI/CD, що робить його інструментом для виявлення вразливостей на ранніх етапах життєвого циклу розробки програмного забезпечення. OWASP ZAP (Zed Attack Proxy) гарантує динамічне тестування безпеки додатків, необхідне для виявлення вразливостей під час роботи інформаційної системи. Його можливості як проксі-перехоплювача дозволяють проводити глибокий аналіз поведінки програми, пропонуючи як активні, так і пасивні параметри сканування. Інструмент безпеки Archery діє як центральна

платформа для агрегування вразливостей, виявлених іншими інструментами та керування ними. Це спрощує відстеження, аналіз і звітування про недоліки безпеки, полегшуючи визначення пріоритетів та ефективне усунення вразливостей.

### 3.4. Розробка методології дослідження кіберзлочинів на основі виявлення аномалій моделлю Ізоляційний Ліс та GPT з урахуванням вразливостей інформаційних систем та даних розвідки про загрози

У цьому розділі пропонуємо методологію дослідження кіберзлочинів для складових інфраструктури інформаційних систем на основі алгоритмів виявлення аномалій ізоляційного лісу та моделей GPT з урахуванням вразливостей інформаційних систем та даних розвідки про загрози. Модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем представлено на Рис 3.3.

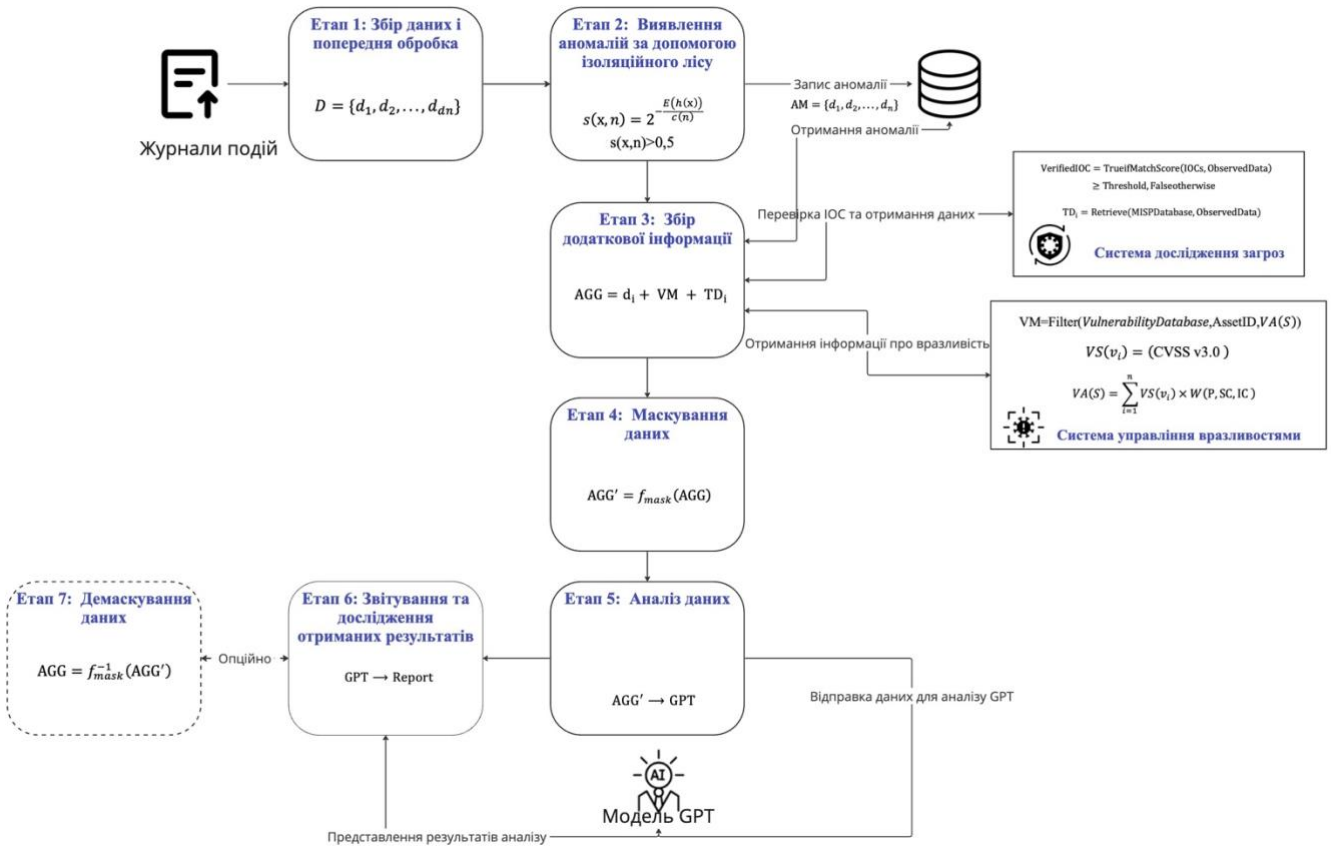



Рис. 3.3. Модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем

Варто зазначити, що вказана модель системи дослідження кібезлочинів не є залежною від версії моделі GPT та побудована з урахуванням принципу постійного

покращення, тому надає можливість командам інформаційної безпеки інтегрувати будь-які зовнішні системи з підтримкою моделі GPT.

Розроблена модель пропонує дослідження кіберзлочинів за наступною методологією.

Етап 1: Збір даних і попередня обробка. Журнали подій централізовано збираються з компонентів різних рівнів інфраструктури інформаційної системи. Це можуть бути журнали мережевого трафіку, журнали доступу до системи, записи транзакцій тощо. Важливо додати, що дані потрібно очистити і попередньо обробити, щоб переконатися, що вони мають відповідний формат для аналізу. Для математичного опису масиву журналів для моделі, особливо у контексті дослідження кіберзлочинів з використанням методів, таких як ізоляційний ліс, потрібно представити журнали у структурованому форматі даних, зазвичай як матрицю. Кожен рядок у матриці представляє окремий запис журналу, а кожен стовпець – окрему характеристику або атрибут, витягнутий з журналів. Нехай D (від англійського Data – дані) є матрицею, що представляє масив журналів. Припустимо, є n записів журналів і m характеристик, витягнутих з кожного запису журналу.

Тоді D представляється як матриця:

$$D = \{d_1, d_2, \ldots, d_{dn}\}, \tag{3.7}$$

Етап 2: Виявлення аномалії за допомогою ізоляційного лісу. Навчання моделі ізоляційного лісу на наборі даних з нормальною поведінкою є першочерговим завданням. Ізоляційний ліс ефективний для виявлення аномалії, після чого він ізолює аномалії. Для навчання моделі потрібно отримати із журналів відповідні записи, які можуть вказувати на аномалії. Це може включати такі методи, як PCA (аналіз основних компонентів) для зменшення розмірності або більш складної розробки функцій. Процес навчання ізоляційного лісу передбачає побудову кількох ізольованих дерев. Математичний опис процесу навчання моделі можна описати таким чином:

1. Ініціалізація: нехай у лісі буде T дерев. Для кожного дерева $t_i$ , де (i =1,2,3…,T), випадковий набір зразків ($S_I$) вибрано з масиву D [92].

2. Рекурсивний поділ у дереві $t_i$:. Вибрано випадкову функцію $f_j$ випадкового значення розділення v між мінімальним і максимальним значеннями $f_j$ у випадковому наборі зразків $S_I$. Розділено на дві підмножини на основі значення розділення в функції $f_j$. Розділення продовжується, доки всі зразки не будуть ізольовані або не буде встановлено певне обмеження глибини [92].

3. Розрахунок довжини шляху: Довжина шляху для зразка у $h(x)$ в кожному дереві $t_i$. Оцінка аномалії для кожного зразка у:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}, \quad (3.8)$$

де s(x,n) – оцінка аномалії вибірки x, E(h(x)) – середня довжина шляху в дереві, а c(n) – середня довжина шляху у випадковому бінарному дереві пошуку.

4. Опрацювання аномалій, виявлених ізоляційним лісом. Для кожної аномалії визначається порогове значення її оцінки, для нормальної події це значення становить від 0 до 0,5. Значення s(x,n)>0,5 свідчить про наявність аномалії та оцінка дуже близька до 1 вказує на точну наявність аномалії. Якщо така аномалія виявлена, аномалії додаються у множину аномалій AM:

$$AM = \{d_1, d_2, \ldots, d_n\}, \quad (3.9)$$

де AM масив аномалій, d - виявлена аномалія та n – кількість аномалій.

Етап 3: Збір додаткової інформації про інформаційну систему та її складові. Запропоновано використовувати додаткові джерела даних, щоб додати контекст до аномалій. Дану інформацію потрібно отримати з системи дослідження загроз та з системи аналізу вразливостей. У випадку, якщо журнал подій з аномалією містить індикатор компрометації систем з вказаних джерел загроз або вразливість, виявлену системами виявлення вразливостей, ці дані потрібно зібрати у один масив даних. Для збору даних з системи дослідження загроз пропонуємо виконати перевірку наявності індикаторів компрометації у системі дослідження загроз:

$$\text{VerifiedIOC} = \text{TrueifMatch}(\text{IOCs}, \text{ObservedData}) \geq \text{Threshold}, \text{False}, \quad (3.10)$$

Та виконати наступну функцію збору індикаторів компрометації при отриманому значенні > 1.

$$\text{TD}_i = \text{Retrieve}(\text{MISPDatabase}, \text{ObservedData}), \quad (3.11)$$

де $TD_i$ представляє функцію для отримання даних про загрози, які можуть включати індикатори компрометації (IOC), і – порядковий номер. Retrieve (отримання) – це операція, що представляє процес запиту інформації до системи MISP. MISPDatabase – база даних MISP. ObservedData підтверджений індикатор компрометації. Наступним кроком є збір даних про вразливості інфраструктури інформаційної системи. Даний крок описаний такою формулою:

$$VM = Filter(VulnerabilityDatabase, AssetID, VA(S)), \quad (3.12)$$

де $VA(S)$ – це загальна оцінка вразливості системи, AssetID – це номер активу, а VulnerabilityDatabase – база даних вразливостей системи Archery.

$$AGG = d_i + VM + TD_i, \quad (3.13)$$

де AGG масив аномалій, , $d_i$ - виявлена аномалія, і – порядковий номер.

Етап 4: Маскування даних. Перед подачею даних журналу (матриця AGG) у модель дослідження кіберзлочинів потрібно застосувати техніку маскування даних. Маскування даних можна представити, як функцію $f_{mask}$, яка приймає вихідну матрицю даних YM і повертає замасковану матрицю:

$$AGG' = f_{mask}(AGG), \quad (3.14)$$

Етап 5: Аналіз даних. Замаскована матриця, дані з джерел загроз та інформація про вразливості інформаційної системи надсилається до GPT моделі для синтезу інформації:

$$AGG' \longrightarrow GPT, \quad (3.15)$$

Етап 6: Звітування та дослідження отриманих результатів. На даному етапі проводиться дослідження отриманих результатів, проведених моделлю GPT:

$$GPT \longrightarrow Report, \quad (3.16)$$

Етап 7 (Опційно): Де-маскування даних. У випадках, коли вихідні дані потрібно отримати для детального аналізу або звітності, слід використати функцію, зворотну функції маскування:

$$AGG = f_{mask}^{-1}(AGG') \quad (3.17)$$

Дії, які модель дослідження кіберзлочинів повинна виконати за визначеною методологією, представлено на Рис. 3.4.

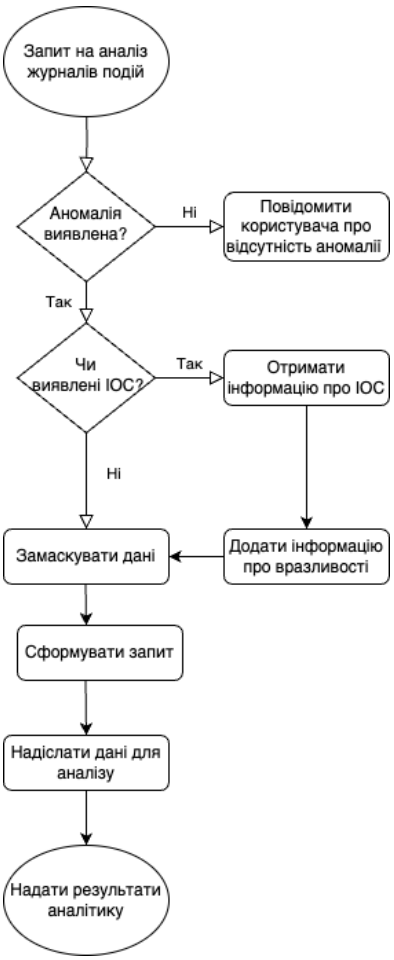



Рис. 3.3. Блок-схема дій

1. Коли аналітик надає журнали подій моделі дослідження кіберзлочинів зібраний за допомогою динамічної системи приманок на основі технології Blockchain, модель активує алгоритм виявлення аномалій. Натренована модель виявляє аномалію та автоматично збирає в один масив даних.

2. Якщо аномалія виявлена, модель запитує дані з системи аналізу вразливостей та отримує інформацію про вразливості сторонніх бібліотек та сканування на вразливості додатку та інфраструктури інформаційної системи.

3. Наступною перевіркою є контроль інших індикаторів компрометації систем, зокрема IP-адрес та доменних імен у системі дослідження загроз.

4. Якщо індикатори компрометації присутні, модель додає інформацію про загрози до масиву даних для аналізу.

5. Наступним кроком є запит даних про вразливості, якщо вразливість виявлена, дані про вразливість збирає модель в окремий масив даних, який вона використовуватиме для аналізу за допомогою GPT.

6. Сформований масив даних маскують, усі IP адреси та доменні імена підміняються на псевдовипадкові.

7. Замаскований масив даних надсилають у модель GPT для аналізу.

8. Після аналізу здійснюють демаскування даних та результати перевірки аналізує аналітик інформаційної безпеки.

У наступному розділі детально описано практичну реалізацію моделі системи дослідження кіберзлочинів та принцип її роботи.

### 3.5. Практична реалізація моделі системи дослідження кіберзлочинів

Позначення, які використовуються для моделювання системи, описані у Таблиці 3.4, а Таблиця 3.5 описує дії системи.

Таблиця 3.4.

Позначення, які використовують для моделювання системи.

| Назва | Позначення |
|---|---|
| Журнали подій - Logs | $\{log_1, log_2, \ldots, log_n\}$ |
| Вразливості - Vulnerabilities | $\{V_1, V_2, \ldots, V_n\}$ |
| Індикатори компрометації (IOC) | $\{IOC_1, IOC_2, \ldots, IOC_n\}$ |
| IP адреси - IP | $\{IP_1, IP_2, \ldots, IP_n\}$ |
| Доменні імена - Domains | $\{d_1, d_2, \ldots, d_n\}$ |
| Аномалія - Anomaly | $\{A_1, A_2, \ldots, A_n\}$ |
| Масив даних - Array | $\{Arr_1, Arr_2, \ldots, Arr_n\}$ |
| Результат - Result | $\{R_1, R_2, \ldots, R_n\}$ |

Таблиця 3.5.

Опис дій системи

| Функція | Опис |
|---|---|
| Отримати (журнали подій) – get (logs) | Модель на базі ізоляційного лісу отримує дані для аналізу алгоритмом |
| Аналіз (журнали подій) – analyse (logs) | Відбувається аналіз журналів подій алгоритмом ізоляційного лісу, для виявлення аномалій |
| Записати (аномалію) – record (anomaly, array) | Система записує аномалій у масив даних, якщо така виявлена. |
| Перевірити наявність вразливостей (вразливості) – check (vulnerabilities) | Відбувається перевірка наявності вразливостей у системі управління вразливостями. |
| Записати (вразливості) – record (vulnerability) | Система записує вразливості в масив даних, якщо така виявлена. |
| Перевірити наявність індикаторів компрометації систем (IOC) – check (IOC) | Відбувається перевірка наявності індикаторів компрометації у системі аналізу загроз. |
| Записати (IOC) – record (IOC) | Система записує індикатор і загрозу в масив даних, якщо такі виявлені. |

Продовження таблиці 3.5.

| | |
|---|---|
| Замаскувати масив – mask (array) | Функція маскування даних підмінює IP адреси та доменні імена на псевдовипадкові. |
| Сформувати запит (масив)– form request (array) | Система формує масив даних для аналізу. |
| Відправити дані для аналізу (масив) – send (array) | Система відправляє масив даних для аналізу до системи з підтримкою GPT. |
| Отримати результат аналізу (result) – receive (result) | Система отримує результат аналізу з системи з підтримкою GPT. |
| Демаскувати масив – demask (array) | Функція демаскування даних змінює псевдовипадкові IP адреси та доменні імена на реальні. |
| Повідомити користувача (результат) -inform (result) | Система відсилає користувачу результати аналізу. |
| Повідомити користувача (аномалія відсутня) - inform (anomaly not detected) | Система повідомляє користувача про відсутність аномалії. |

Для опису роботи системи дослідження кіберзлочинів розроблено Алгоритм 1.

Алгоритм 1. Дослідження кіберзлочинів

Функція get(log) отримує масив журналів подій $\{log_1, log_2, \ldots, log_n\}$ у форматі .log. Отримана інформація обробляється моделлю ізоляційний ліс для виявлення аномалії. За умови виявлення аномалії система записує аномалії за допомогою функції record (anomaly) у базу даних:

$$\text{Ізоляційний Ліс (analyse (logs))} \rightarrow \text{record (anomaly)}$$

$$\rightarrow \{Arr_1, Arr_2, \ldots, Arr_n\}\ else\ \text{inform (anomaly not detected)}$$

Далі функція check (vulnerabilities) запитує інформацію у системи виявлення вразливостей про наявність вразливих компонентів інформаційної системи:

$$if\ \text{check (vulnerabilities)} = True \rightarrow then\ Record\_in\_Arr: \{V_1, V_2, \ldots, V_n\}\ else$$

$$\rightarrow \quad No\ action$$

Функція check (IOC) перевіряє про наявність індикаторів компроментації систем у системі дослідження загроз:

$$if\ \text{check (IOC)} = True \rightarrow then\ Record\_in\_Arr: \{IOC_1, IOC_2, \ldots, IOC_n\}\ else$$

$$\rightarrow \quad No\ action$$

Функція form(array) формує масив даних для аналізу системою GPT та сформований масив даних маскується функцією mask (array).

$$\text{form request(array)} \rightarrow \text{mask (array)} \rightarrow array'$$

Сформований масив даних надсилають до системи дослідження на базі GPT за допомогою API з’єднання за допомогою функції send (array).

$$send(array') \rightarrow GPT$$

Результати аналізу отримуються функцією receive (result), наступним етапом дані демаскуються demask ($array'$) та передаються аналітику функцією inform (result). У такий спосіб система обробляє дані та надає підказку аналітику про можливу першопричину виникнення інциденту або ж кіберзлочину. Для описаної системи була створена діаграма послідовності UML, зображена на Рис. 3.5. Вона окреслює взаємодію та процеси в системі, залучаючи аналітика, систему дослідження кіберзлочинів та зовнішню модель GPT. Ця діаграма забезпечує візуальне представлення послідовності операцій, включаючи прийом журналу,

дослідження, перевірку вразливостей і загроз, маскування даних, зв’язки із зовнішніми системами та остаточний звіт про результати.

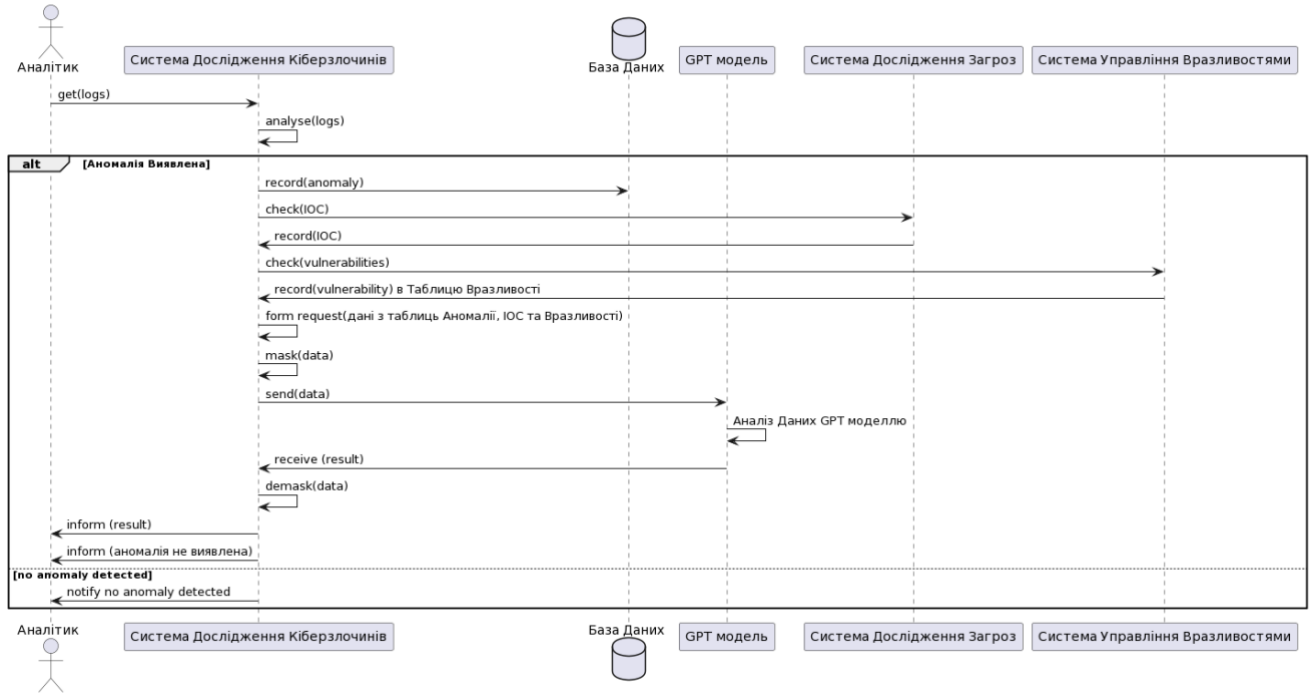



Рис. 3.5. Діаграма послідовності

Програмна архітектура моделі подана за допомогою C&C вигляду на Рис. 3.4, що вказує на елементи системи, а саме її сервіси та з’єднання між сервісами системи. Варто зазначити, що для системи було обрано мікросервісний тип архітектури, оскільки система реалізована, базуючись на даному типі архітектури, є завадостійкою, забезпечує можливість масштабування та гнучкого вибору технологій. Дані характеристики стали важливим фактором при виборі типу архітектури системи, адже мікросервіси дозволяють легко масштабувати окремі компоненти програми, тому що кожну службу можна масштабувати незалежно на основі попиту та кожен мікросервіс можна створити за допомогою різних технологій (мов програмування, баз даних тощо), що дозволяє використовувати різні технології для потреб кожного конкретного сервісу, до прикладу: аналізу вразливостей, збору індикаторів компрометації чи GPT моделі. Тому систему дослідження кіберзлочинів реалізовано за допомогою сервісів у контейнеризованому середовищі.

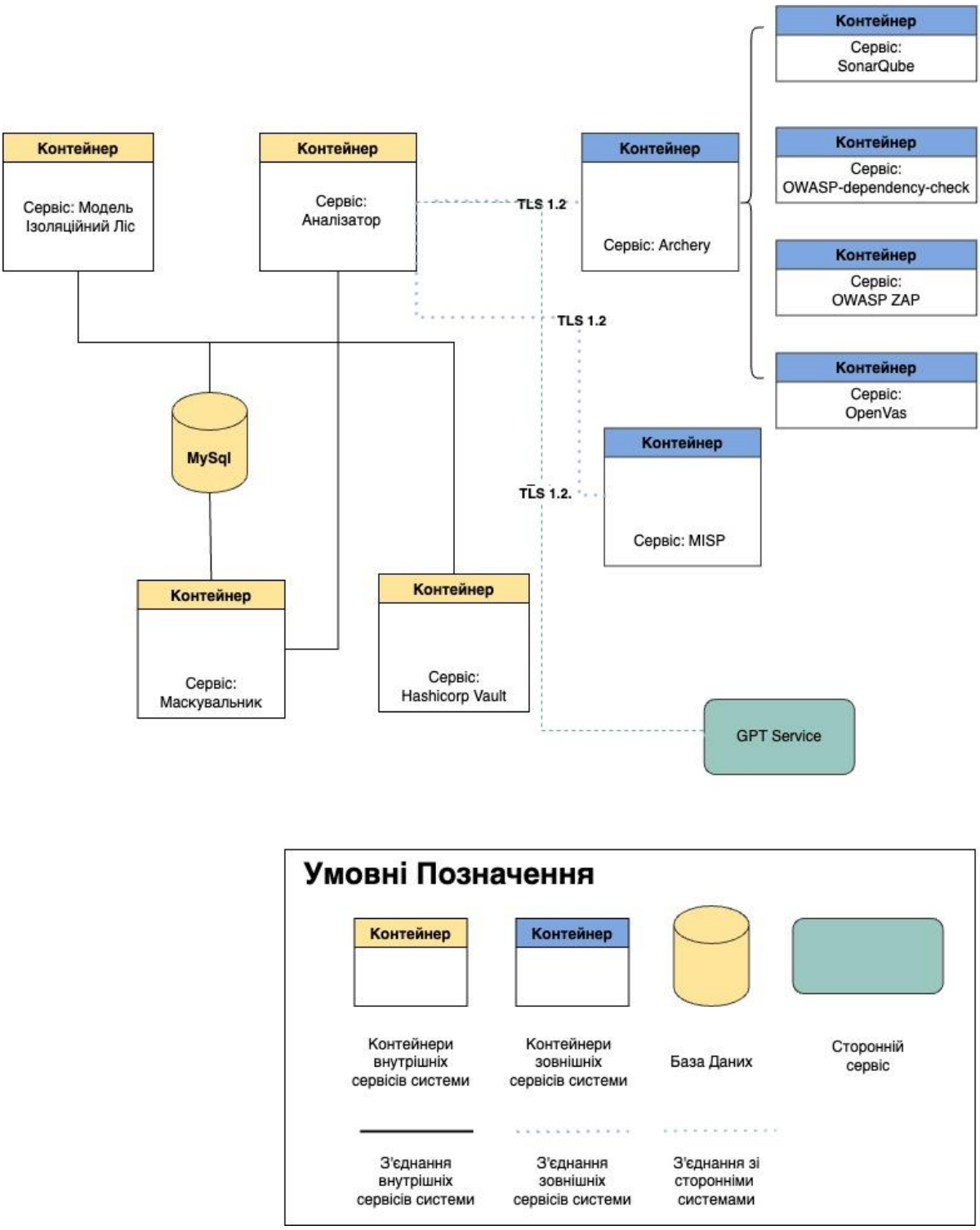



Рис. 3.4. Діаграма компонентів (C&C)

Компоненти системи описані у Таблиці 3.6.

Таблиця 3.6.

Компоненти системи

| Компонент | Опис |
|---|---|
| Сервіс: Аналізатор | Сервіс аналізатор реалізований на мові програмування Python та є основним сервісом моделі. Даний сервіс виконує наступні функції: збір інформації про аномалію з сервісу, вразливості системи з сервісу Archery та індикатори компрометації системи з сервісу MISP. Цей сервіс формує запит до моделі GPT. Відправляє дані на маскування та замасковані дані надсилає моделі GPT, у разі отримання відповіді повідомляє аналітика про результати дослідження або про відсутність аномалії. |
| Сервіс: Модель ізоляційний ліс | Сервіс, реалізований на базі моделі ізоляційного лісу на мові програмування Python, попередньо є натренованою моделлю та отримує дані для аналізу. Результатом виконання є виявлена та задокументована аномалія. |
| Сервіс: Маскувальник | Сервіс реалізований на мові програмування Python, що маскує сформовані для аналізу дані та демаскує отримані дані від моделі GPT. |
| Сервіс: Hashicorp Vault | Сервіс Hashicorp Vault використовують для збереження ключів API. |
| Сервіс: Arhery | Сервіс Archery агрегує дані з систем виявлення вразливостей та є реалізацією, описаним у розділі 3.2 компонентом управління вразливостями. |

Продовження таблиці 3.6.

| | |
|---|---|
| Сервіс: MISP | Сервіс MISP агрегує дані з джерел загроз та є реалізацією, описаної у розділі 3.1 системою дослідження загроз. |
| Сервіс: GTP Service | Зовнішня GPT модель інтегрована у систему дослідження кіберзлочинів. |
| База Даних - MySql | База даних, що реалізована на MySql. У базу даних записують інформацію про виявлені вразливості, аномалії та індикатори компрометації систем. |

Алгоритми системи було розроблено використовуючи мову програмування Python завдяки обширній екосистемі бібліотек, яка містить такі інструменти, як Pandas для маніпулювання даними, Scikit-learn для машинного навчання та Matplotlib для візуалізації, які спрощують реалізацію складних алгоритмів штучного інтелекту. Також Python забезпечує інтеграцію з іншими технологіями, що є необхідним для реалізації запропонованої системи.

**Висновки до розділу 3**

У третьому розділі дисертаційної роботи розроблено та запропоновано методологію дослідження кіберзлочинів з використанням алгоритмів ізоляційного лісу, GPT та підходу DevSecOps та модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем.

Проведене дослідження дозволяє виділити наступне:

1. Представлена методологія дослідження кіберзлочинів заснована на використанні системи дослідження загроз, підході DevSecOps, моделі ізоляційний ліс та моделі GPT. Ця методологія може забезпечити комплексний аналіз кіберзлочинів та швидке визначення першопричини виникнення інцидентів для складових інфраструктури інформаційних систем.

2. Запропонована модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем, що використовує методологію визначення аномалій шляхом навчання ізоляційного лісу нормальній поведінці інформаційної системи, що дає можливість адаптувати її під будь-яку інформаційну систему. Цей підхід покращує класифікацію шкідливих запитів і зменшує кількість фальшивих спрацювань. Система унікально ідентифікує кожен запит та визначає, чи містить він аномалію. До кожної виявленої аномалії здійснюють перевірку наявності індикаторів компрометації систем, що дозволяє точно визначити загрозу та дає можливість дізнатись публічну інформацію про зловмисника чи групу зловмисників. Також система інтегрується з DevSecOps рішеннями для аналізу наявності вразливостей та за допомогою вагових коефіцієнтів забезпечує їх точніший аналіз і надає можливість аналітику оцінити їх серйозність для даної системи.

3. Модель системи дослідження кіберзлочинів розроблена з урахуванням принципів “Безпека за замовчуванням” та “Безпека за дизайном”, що виражено застосуванням алгоритму TLS 1.2 для передачі даних та впровадженням функціоналу маскування даних для зібраної інформації з різних систем. Функцію дослідження, що може допомогти аналітику під час опрацювання кіберзлочину, виконує GPT модель, результати з якої хоч і не повинні бути ключовими для прийняття рішення, проте мають суттєво скоротити час дослідження кіберзлочинів та визначення першопричини їх виникнення.

# РОЗДІЛ 4. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ

## 4.1. Підготовка вразливого середовища для тестування моделі системи дослідження кіберзлочинів

На сучасному етапі розвитку інформаційних систем кібербезпека зосереджена переважно на захисті, який виявляє та запобігає кібератакам. Однак набагато важливіше регулярно перевіряти стан безпеки організації, щоб посилити захист кібербезпеки, оскільки IT-середовище стає складнішим і конкурентоспроможним [71].

У цьому розділі представлено результати проведеної симуляції кібератак на підготовлене вразливе середовище, результати симуляції надано системі дослідження кіберзлочинів для аналізу. Середовище для генерування журналів подій складається з системи приманок на основі технології Blockchain, основними вузлами яких є сервіси Nginx, які використовуються як реверсивні проксі сервіси та контейнери вразливого застосунку OWASP JuiceShop. Схематично дане середовище представлено на Рис. 4.1.

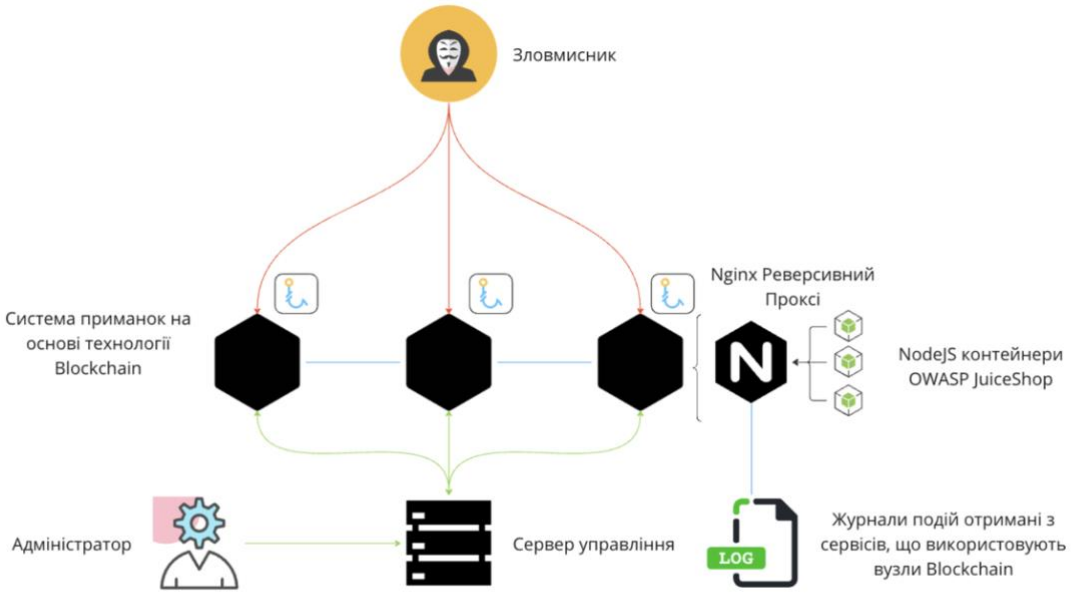



Рис. 4.1. Середовище для симуляції атак

Для експериментального аналізу проведено симуляції наступних кібератак: атака з використанням автоматичних інструментів сканування та експлуатації вразливостей, атака Directory Traversal, ін’єкції та спроби порушення логіки програми.

Використання таких інструментів як Nessus для сканування вразливостей є заходом для виявлення та усунення слабких місць у мережі чи системі. Ці інструменти призначені для виявлення широкого спектру вразливостей, починаючи від недоліків програмного забезпечення та закінчуючи неправильними конфігураціями. Основною перевагою використання цих інструментів є їх комплексний характер, вони надають детальну інформацію про потенційні прогалини в безпеці [94]. Проте зловмисники також можуть використовувати дані рішення, викриваючи вразливості в цілях подальших атак на вибрану систему. Виявлення даного типу атак та його своєчасне попередження може запобігти подальшим інцидентам інформаційної безпеки.

Симуляції ін’єкційних атак, наприклад ін’єкцій SQL і XSS, пропонують оцінку в реальному часі того, наскільки добре системи організації можуть протистояти поширеним і складним методам атак. Ці симуляції імітують дії кіберзлочинців, надаючи реалістичну перевірку заходів безпеки [95]. Перевага тут полягає в можливості оцінити ефективність протоколів безпеки та визначити конкретні області, де захист може потребувати посилення. Цей підхід до тестування безпеки допомагає точно налаштувати захист від поширених векторів атак.

Атака Directory Traversal, також відома як Path Traversal, спрямована на доступ до файлів і каталогів, які зберігаються за межами кореневої папки веб-сайту. Маніпулюючи змінними, які посилаються на файли за допомогою послідовностей крапка-крапка-слеш (../) і подібних конструкцій, зловмисник може переміститися вгору по дереву каталогів від кореня веб-сервера, щоб отримати доступ до довільних файлів або каталогів.

Визначення спроби порушення логіки програми користувачами має вирішальне значення для виявлення внутрішніх загроз або зламаних облікових

записів. Перевагою цього підходу є його акцент на людському елементі кібербезпеки. Відстежуючи аномалії в поведінці користувачів, наприклад незвичайний час входу або несподіваний доступ до даних, організації можуть швидко виявляти та досліджувати потенційні інциденти безпеки [96]. Цей метод особливо ефективний для виявлення загроз, які можуть бути упущені автоматизованими системами, включно з непомітними діями, які можуть вказувати на злам або зловмисну внутрішню діяльність.

У наступних розділах описано проведені експерименти, результати яких були зафіксовані у журнали подій, для подальшого аналізу системою дослідження кіберзлочинів.

### 4.2. Тренування моделі ізоляційний ліс

У зв’язку з використанням журналів подій Nginx для успішного виявлення аномалії під час експерименту потрібно провести навчання моделі ізольованого лісу на журналах NGINX. Тренування включає кілька етапів, починаючи від розуміння та попередньої обробки даних до навчання моделі. Для тренування моделі було використано test_nginx.log файл, що містив 50210 запитів, включно з аномаліями та нормальною поведінкою.

Журнали подій Nginx містили таку інформацію, як віддалені IP-адреси, мітки часу, методи запиту HTTP, коди статусу відповіді та агенти користувача. Кожне з цих полів потенційно може допомогти у виявленні аномалій (наприклад, незвичайні шаблони доступу або рівень помилок). Оскільки ізоляційний ліс працює лише з числовими форматами даних, інформація з журналів подій була отримана та для роботи моделі конвертована у числовий формат даних за такими принципами [97]:

- IP-адреса: конвертована в числовий формат.
- Мітка часу: перетворена на об’єкт datetime та визначено числові функції.
- Метод HTTP: отримано з рядка запиту.
- Запитана URL-адреса: для аналізу URL-адреси її було розбито на

компоненти: протокол, домен, шлях та рядок запиту. Для цього використано бібліотеку urllib Python. Для URL-адреси визначено наступні числові характеристики:

- Довжина URL-адреси: аномально довгі URL-адреси можуть свідчити про підозрілу діяльність, наприклад, впровадження SQL-запиту або атаки з проходженням шляху не є властивими для нормальної діяльності користувача.
- Глибина шляху: обчислено глибину запиту, підрахувавши кількість похилих рисок у шляху URL-адреси. Для кожного додатку можна визначити максимальну глибину запиту, значення більше даної глибини є аномальним.
- Спеціальні символи: наявність певних спеціальних символів (наприклад, “%”, “..” або незвичне кодування) може свідчити про спроби використання вразливостей.

- Версія HTTP: отримано з рядка запиту.
- Код статусу: вказує на результат запиту (наприклад, 404 – не знайдено, 200 – успішно).
- Розмір відповіді: розмір відповіді в байтах.
- Агент користувача: для цього поля було визначено характеристику бота,

використовуючи двійковий прапор (0 або 1), що вказує, чи належить агент користувача боту.

Підготовлені для моделі дані було завантажено model_training.cvs файл, для якого було визначено атрибути аномальної поведінки. Для підготовлених даних ініціалізовано модель ізоляційного лісу. Ізоляційний ліс призначав оцінку аномалії кожному спостереженню з підготовлених даних. Цю оцінку використано, щоб визначити, чи є спостереження аномалією. Останнім етапом стало збереження моделі для подальшого використання, що було забезпечено шляхом використання бібліотеки joblib.

Отже, навчання моделі ізоляційного лісу на журналах Nginx полягало в перетворенні необроблених даних журналу у формат, який може опрацювати

алгоритм, у навчанні моделі виявляти аномалії, а потім інтерпретувати їх у контексті трафіку та активності веб-сервера.

### 4.3. Збір додаткової інформації

Збір додаткової інформації відбувається з системи управління вразливостями та системи дослідження загроз. Попередньо для аплікації JuiceShop було проведено сканування на вразливості.

Сканування інфраструктури відбувалось за допомогою застосунку OpenVas, під час сканування було виявлено одну вразливість низького рівня. Статичне сканування реалізовано використанням OWASP Dependency Check, що виявило 12 критичних вразливостей, 26 вразливостей високого рівня та 29 вразливостей середнього рівня у сторонніх бібліотеках, які використовує інформаційна система. Динамічне сканування проводилось, використовуючи застосунок OWASP ZAP, під час якого було виявлено 2 середнього та 5 низького рівнів. Сканування були завантажені у систему управління вразливостями Archery. Результати сканування на вразливості представлено на Рис. 4.2.

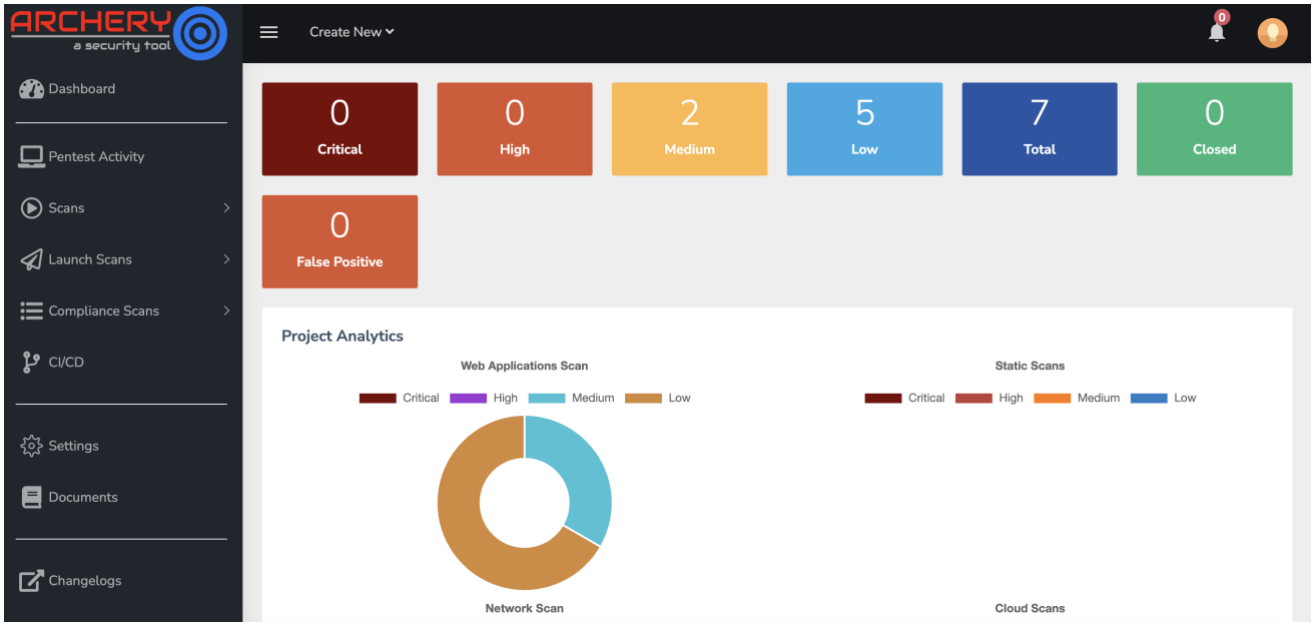



Рис. 4.2. Результати сканування на вразливості

Усі згадані системи сканування реалізовані з урахуванням принципу мікро-сервісної архітектури та впроваджені у реалізацію системи дослідження кіберзлочинів за допомогою Docker-контейнерів для забезпечення вільного масштабування системи. Безпечну комунікацію між системами забезпечують за

допомогою алгоритму TLS 1.2. Інтеграція системи управління вразливостями з системою дослідження кіберзлочинів відбувається за допомогою API з'єднання. Для безпечного збереження облікових даних сервісного користувача, що використовують для API-з'єднання, було впроваджено сховище ключів Hashicorp Vault.

Для виявленої аномалії реалізовано алгоритм для отримання IP-адрес із журналів NGINX і перевірки їх у системі дослідження загроз, що реалізується на Docker-контейнері MISP. Припускається, що IP-адреси в журналах відповідають стандартному формату IPv4 із використанням регулярного виразу для їх вилучення. Цей регулярний вираз може потребувати коригування, якщо журнали мають інший формат IP-адреси. Взаємодія з API MISP використовує базову структуру запиту HTTP POST. Запити API налаштовано відповідно до вимог MISP. Крім того, облікові дані сервісного користувача зберігають у сховищі ключів Hashicorp Vault.

## 4.4. Експериментальне проведення атак

У цьому підрозділі детально описано етапи експериментального проведення атак на досліджувану систему. Він представляє проведення атак сканування, ін'єкційних атак, Directory Traversal та атак порушення логіки програм.

### 4.4.1. Експериментальне проведення атаки з використанням автоматичних інструментів сканування

Сканування вразливостей відбувалося з використанням Nessus. Даний застосунок є традиційним методом, який використовують для виявлення вразливостей безпеки в мережевих системах, проте також виконує низку автоматизованих атак [98]. Атака з використанням автоматичних інструментів сканування схематично представлена на Рис 4.3.

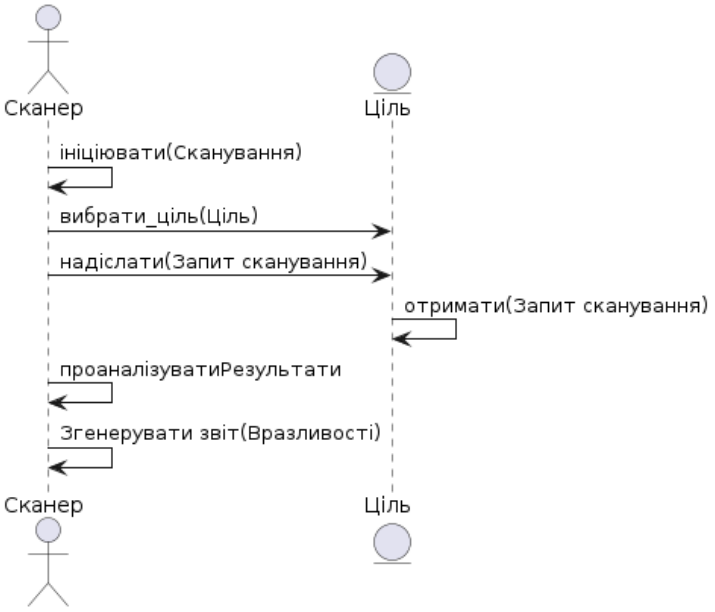



Рис. 4.3. Атака з використанням автоматичних інструментів сканування

Експеримент атаки з використанням автоматичних інструментів сканування містить:

1. Початок сканування: система сканування Сканер починає процес сканування, визначаючи відкриті порти та сервіси. Це представлено як Сканер, що ініціює функцію сканування.

2. Вибір цілі: Сканер вибирає цільовий хост, ціль, для оцінки вразливості. Це описується як Сканер, що вибирає Ціль.

3. Надсилання запитів на сканування: Сканер надсилає низку запитів до Цілі. Ця дія позначається як функція надіслати Запит Сканування. Ці запити призначені для перевірки різних аспектів безпеки Цілі.

4. Цільовий хост сканується: Ціль отримує запити на сканування.

5. Аналіз відповідей: Сканер аналізує відповіді від Цілі, щоб виявити потенційні вразливості.

6. Звіт про вразливості: Сканер складає звіт із детальним описом виявлених вразливостей і потенційних ризиків, представлений.

Основною метою сканування рішеннями Nessus є виявлення слабких місць безпеки в цільовій системі. Дану атаку зловмисник може використовувати для подальшої експлуатації виявлених вразливостей. Результатом сканування протестованої системи на базі OWASP JuiceShop є отриманий звіт Nessus про

проведення сканування. Результати записані у журнали подій Nginx для подальшого аналізу.

### 4.4.2. Експериментальне проведення ін’єкційної атаки

Ін’єкційні атаки, такі як ін’єкція SQL – це зловмисні методи, які використовують зловмисники для використання вразливостей у програмах шляхом ін’єкції несанкціонованого коду в запити чи команди. Основна мета ін’єкційних атак полягає в тому, щоб отримати несанкціонований доступ або вчинити несанкціоновані дії з цільовою програмою [99]. Для тестування ін’єкційних атак використовували програму-тестувальник BurpSuite. Ін’єкційна атака сканування схематично представлена на Рис 4.4.

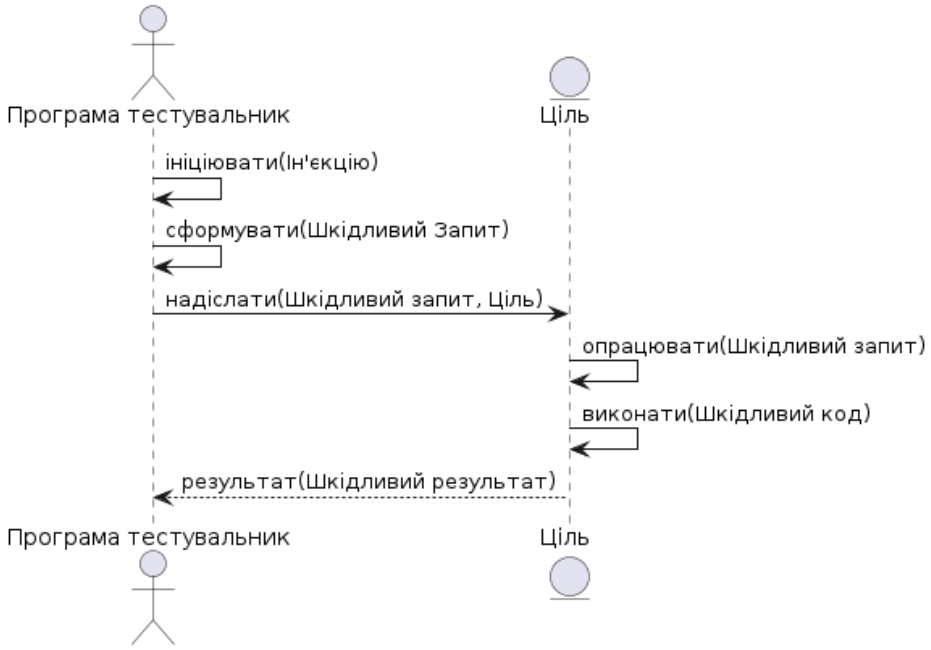



Рис. 4.4. Ін’єкційна атака

Для експерименту атака розгортається в такій послідовності:

1. Програма тестувальник починає процес. На схемі це представлено як Програма тестувальник, що виконує функцію ініціювати (ін’єкцію).

2. Створення шкідливого запиту: тестувальник створює шкідливий запит, який призначений для використання вразливостей цільової програми.

3. Надсилання шкідливого запиту цілі: створений шкідливий запит надсилають в цільову програму.

4. Програма обробляє вхідні дані: ціль обробляє отримані вхідні дані, не знаючи про зловмисні дії запиту.

5. Виконання шкідливого коду: зловмисне введення викликає шкідливі дії ціллю, як-от неавторизований доступ до даних або їх модифікація.

Основною метою атаки SQL є отримання визначення схеми БД та спроби авторизуватись у системі з правами адміністратора. Атака XSS реалізована для впровадження стороннього змісту у інформаційну систему.

### 4.4.3. Експериментальне проведення Directory Traversal атаки

Метою атаки Directory Traversal є доступ до файлів і каталогів, які зберігаються за межами передбачених доступних каталогів веб-сервера. Цей тип атаки використовує вразливі місця у веб-додатку (або веб-сервері), які не можуть належним чином очистити введені користувачем шляхи до файлів. Для симуляції даного типу атаки використовують застосунок Dirbuster. Схематично експериментальне проведення Directory Traversal атаки представлено на Рис. 4.5.

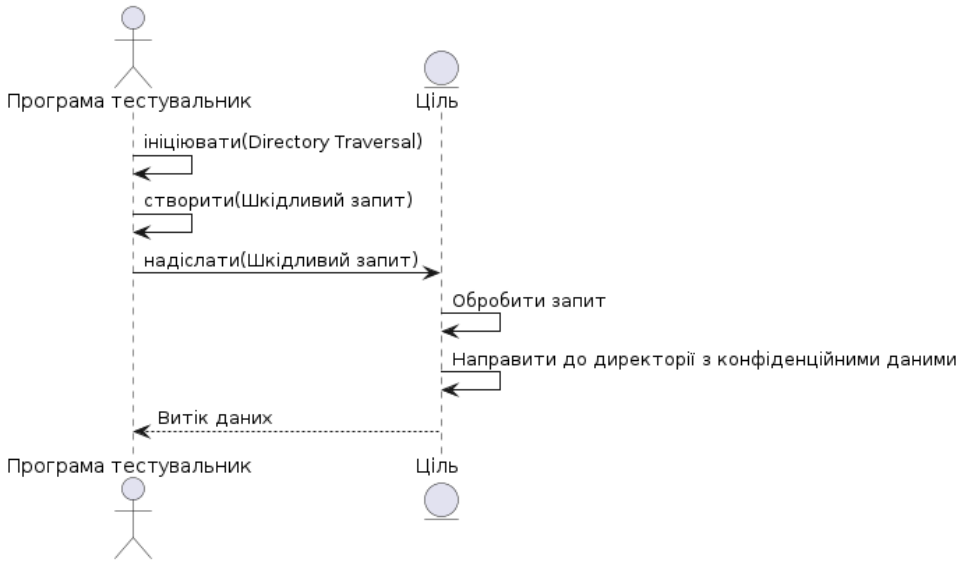



Рис. 4.5. Атака Directory Traversal

Для експерименту, атака розгортається в такій послідовності:

1. Програма Тестувальник починає процес. На схемі це представлено як Програма Тестувальник, що виконує зловмисний запит, який містить послідовності обходу каталогу (наприклад, '../').

2. Надсилання зловмисного запиту: програма Тестувальник надсилає запит на цільовий сервер. Сервер обробляє запит: цільовий сервер обробляє запит, не дезінфікуючи вхідні дані належним чином.

3. Перехід до каталогу з обмеженим доступом: сервер отримує доступ до каталогу або файлу за межами призначеного каталогу в результаті Directory Traversal.

4. Компрометація даних: тестувальтник отримує доступ до конфіденційних файлів або каталогів, що потенційно може призвести до витоку даних.

Результати реалізації експерименту зазначені у журналах подій.

### 4.4.4. Експериментальне проведення спроби порушення логіки програми

У цьому експерименті симулюються дії користувача, зумовлені на порушення логіки програми. Це включає введення даних, для обробки яких програма не призначена, використання логічних недоліків для спроби порушення звичайної роботи програми. Схематично експериментальне проведення спроби порушення логіки програми показано на Рис. 4.6.

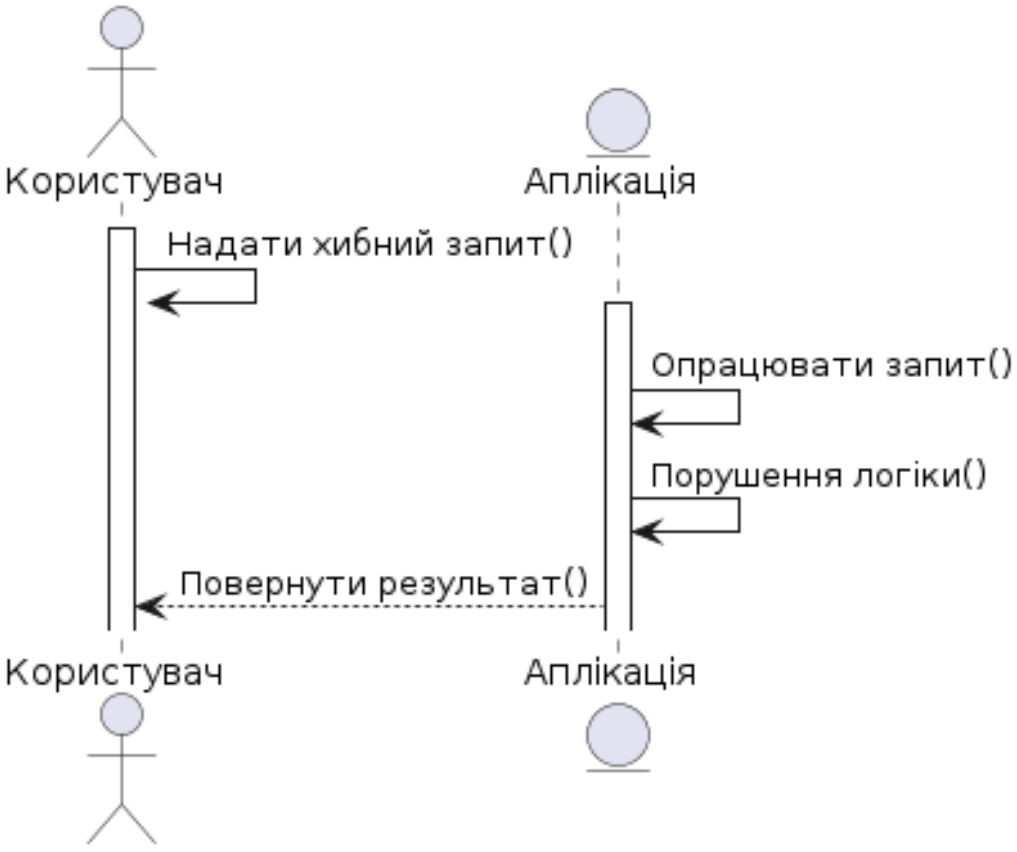



Рис. 4.6. Спроба порушення логіки програми

Для експерименту атака розгортається в такій послідовності:

1. Користувач починає з введення запиту, який використовує відомі логічні недоліки в програмі.

2. Програма обробляє отриманий запит.

3. Програма не виявляє аномалію, та логіка роботи порушується.

4. У відповідь на порушення програма опрацьовує запит та повертає результат запиту.

У випадку з аплікацією тестування використовується програма тестувальник Burp Suite, за допомогою якої симулюється спроба зменшення вартості товару користувачем. Результати записані у журнали подій Nginx для подальшого аналізу.

### 4.5. Аналіз результатів

Цей підрозділ представляє ретельний аналіз роботи запропонованої моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем. У ньому описано аналіз ефективності системи дослідження кіберзлочинів, використовуючи запропоновану модель. Також отримані результати порівняно з системою дослідження та моніторингу подій з відкритим кодом Wazuh та аналізом виконаним спеціалістами з інформаційної безпеки.

Для дослідження журналів подій було використано традиційну SIEM-систему Wazuh та залучено двох аналітиків інформаційної безпеки, для яких виміряно час виявлення події SIEM-системою Wazuh та сумарний час дослідження кіберзлочинів аналітиками інформаційної безпеки. Вимірювання часу виконується за допомогою Google Stopwatch.

#### 4.5.1. Аналіз швидкості виявлення та дослідження атак з використанням автоматичних інструментів сканування

Для цього дослідження використовували Nessus Community Edition, що симулював атаку зі сканування із стандартним набором правил. Для експерименту використовувались записи журналів подій, згенеровані під час автоматичного сканування на вразливості з допомогою Nessus Community Edition та проведено 10 експериментів з 10 наборами даних, що містять 100 однакових записів журналів

подій, кожен з яких містить принаймні одну аномалію. Дослідження проводилось щогодини протягом 10 годин. Як показано на Рис. 4.6, середній час виявлення аномалій, визначений під час даних експериментів, становить 5.1 секунда комплексною системою дослідження кіберзлочинів та середній час виявлення події інформаційної безпеки SIEM системою Wazuh – 13.1 секунда. Тому система дослідження кіберзлочинів є до 61% швидшою за SIEM-систему Wazuh для виявлення кібезлочинів. Порівняння часу виявлення аномалій представлено на Рис. 4.7.

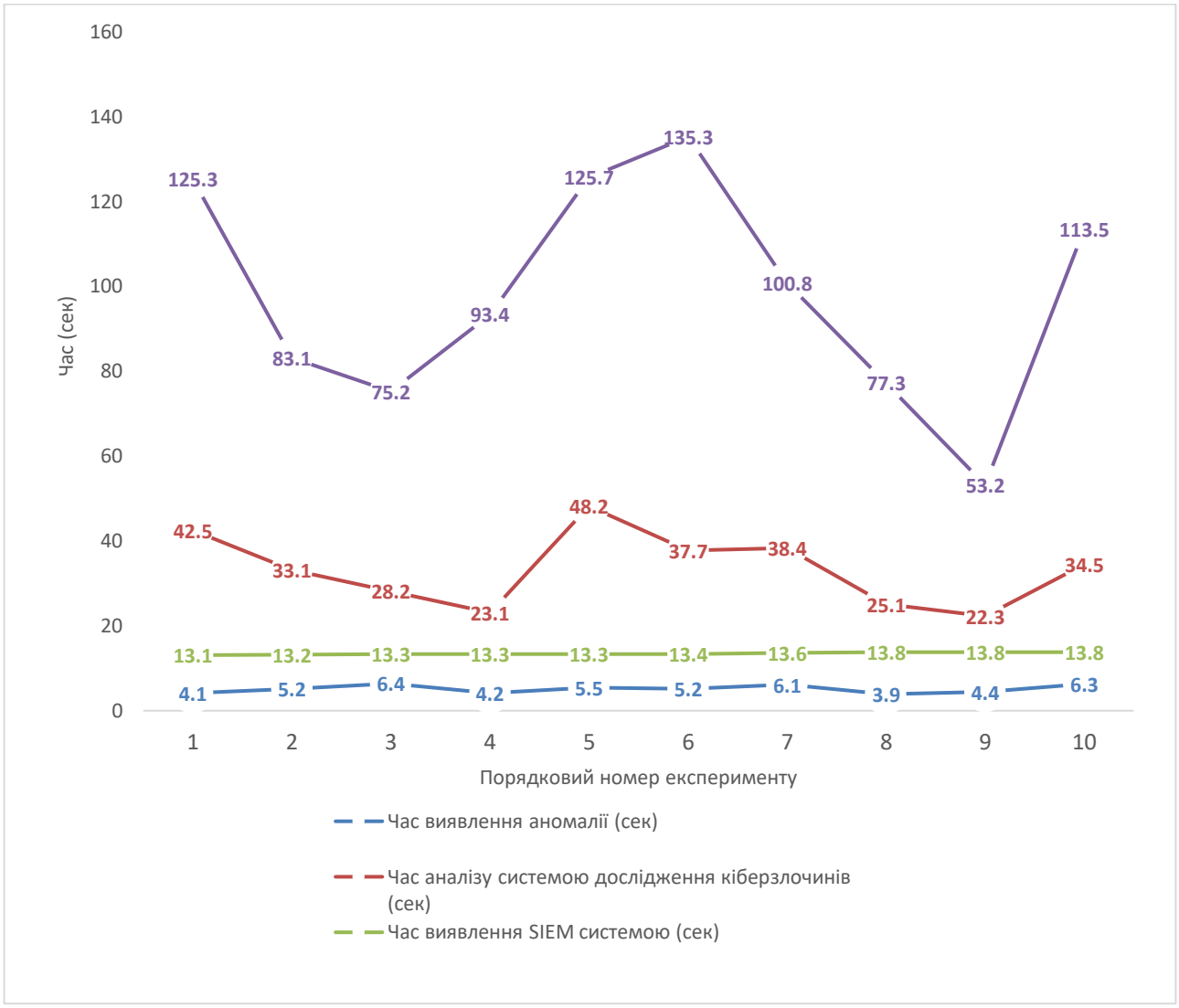



Рис. 4.7. Порівняння часу аналізу атаки сканування комплексною системою дослідження кіберзлочинів та традиційним підходом

Наступний показник, який беруть до уваги, – це час аналізу кіберзлочинів. Середній час аналізу кіберзлочинів спеціалістами інформаційної безпеки з

використанням даних SIEM-системи Wazuh (даний час не враховує звітування) становить 98.28 секунд та 33.31 секунд відповідно системою дослідження кіберзлочинів (даний час враховує отримання детального звіту). Експериментально підтверджено, що система дослідження кібезлочинів може підвищити швидкість дослідження кіберзлочинів приблизно до 66%, з огляду на критерій швидкості дослідження кіберзлочину.

Аналізуючи отримані результати, було встановлено, що система дослідження кіберзлочинів для складових інфраструктури інформаційних систем до 66% швидше виявляє аномалії, що може зосередити увагу аналітиків на потенційну атаку. Також система дослідження кібезлочинів може підвищити ефективність дослідження кіберзлочинів приблизно до 61%.

### 4.5.2. Аналіз швидкості виявлення та дослідження ін’єкційних атак

Для вимірювання швидкодії використовували журнали подій, згенеровані під час виконання атаки ін’єкції, та проведено 5 експериментів. Під час кожного експерименту використовували по 5 журналів подій, кожен з яких містив по 100 записів. Дослідження проводили протягом 5 годин по одному експерименту щогодини. Проведені експерименти визначають, що показник часу, затраченого на виявлення аномалій, становить 5.66 секунд комплексною системою дослідження кіберзлочинів та SIEM-системою Wazuh – 11.02 секунд. Тому встановлено, що система дослідження кіберзлочинів є до 48% швидшою за SIEM-систему Wazuh для виявлення кібезлочинів.

Середній час дослідження кіберзлочинів спеціалістами інформаційної безпеки з використанням даних SIEM-системи Wazuh становив 98.52 секунд та 19.14 секунд відповідно системою дослідження кіберзлочинів. У даному випадку впровадження комплексної системи дослідження кіберзлочинів з використанням GPT з часом аналізу 19.14 секунд може скоротити час аналізу журналів подій на приблизно до 80% у відсотковому відношенні порівняно з SIEM-системою. Результати дослідження представлено на Рис 4.8.

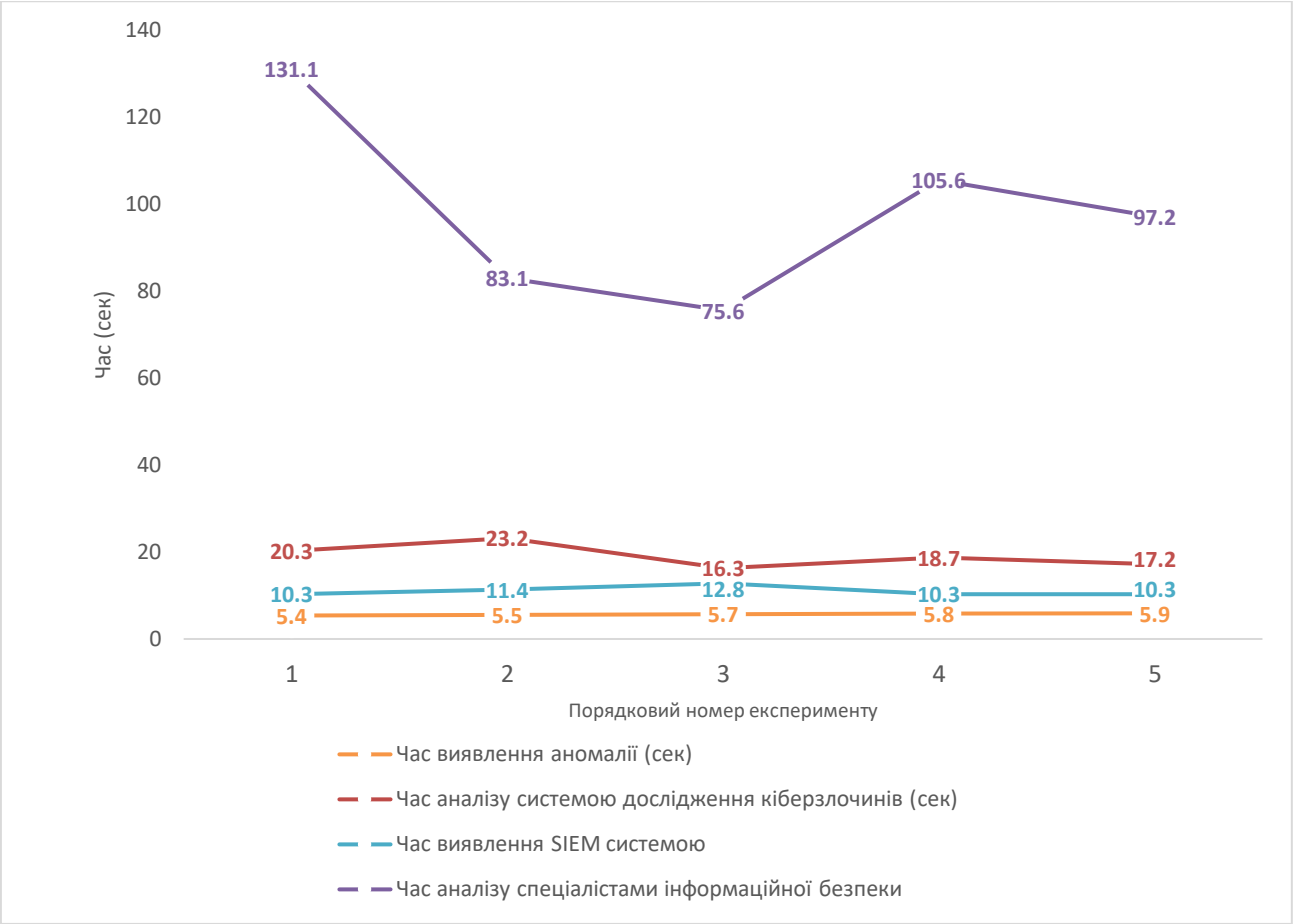



Рис. 4.8. Порівняння часу аналізу ін'єкційних атак комплексною системою дослідження кіберзлочинів та традиційним підходом

Результати проведеного експерименту визначають, що система дослідження кіберзлочинів швидше виявляє події інформаційної безпеки та може зменшити час дослідження журналів подій до 80% у відсотковому відношенні порівняно з SIEM-системою, не зменшуючи при цьому ефективність.

### 4.5.3. Аналіз швидкості виявлення та дослідження Directory Traversal

Для вимірювання швидкодії використовували журнали подій, згенеровані під час виконання атак Directory Traversal за допомогою Dirbuster, та проведено 5 експериментів. Під час кожного експерименту використовували по 5 журналів подій, кожен з яких містив по 100 записів. Дослідження проводили протягом 5 годин по одному експерименту щогодини. Середній час виявлення аномалії становить 6.68 секунд для комплексної системи дослідження кіберзлочинів та час виявлення події інформаційної безпеки SIEM-системою – 7.44 секунд. Тому

система дослідження кіберзлочинів може скоротити час виявлення подій до 7.80% у відсотковому відношенні. Найнижчий показник часу, затраченого на виявлення аномалії, становить 6.1 секунд для системи дослідження кіберзлочинів та 7.4 для SIEM-системи Wazuh. Тому, як бачимо, система дослідження кіберзлочинів є до 17 % швидшою за SIEM-систему для виявлення кіберзлочинів, якщо враховувати час виявлення як критерій ефективності. Наступний показник, визначений для оцінки ефективності, є час аналізу кіберзлочинів. Середній час аналізу кіберзлочинів спеціалістами інформаційної безпеки з використанням даних SIEM-системи є 23.56 секунд та 14.54 секунд відповідно системою дослідження кіберзлочинів. У цьому випадку впровадження комплексної системи дослідження кіберзлочинів з використанням GPT з часом аналізу 14,54 секунди може скоротити час аналізу журналів подій до 38% у відсотковому відношенні порівняно з SIEM-системою. На Рис. 4.9 представлено порівняння часу аналізу атаки Directory Traversal комплексною системою дослідження кіберзлочинів та традиційним підходом.

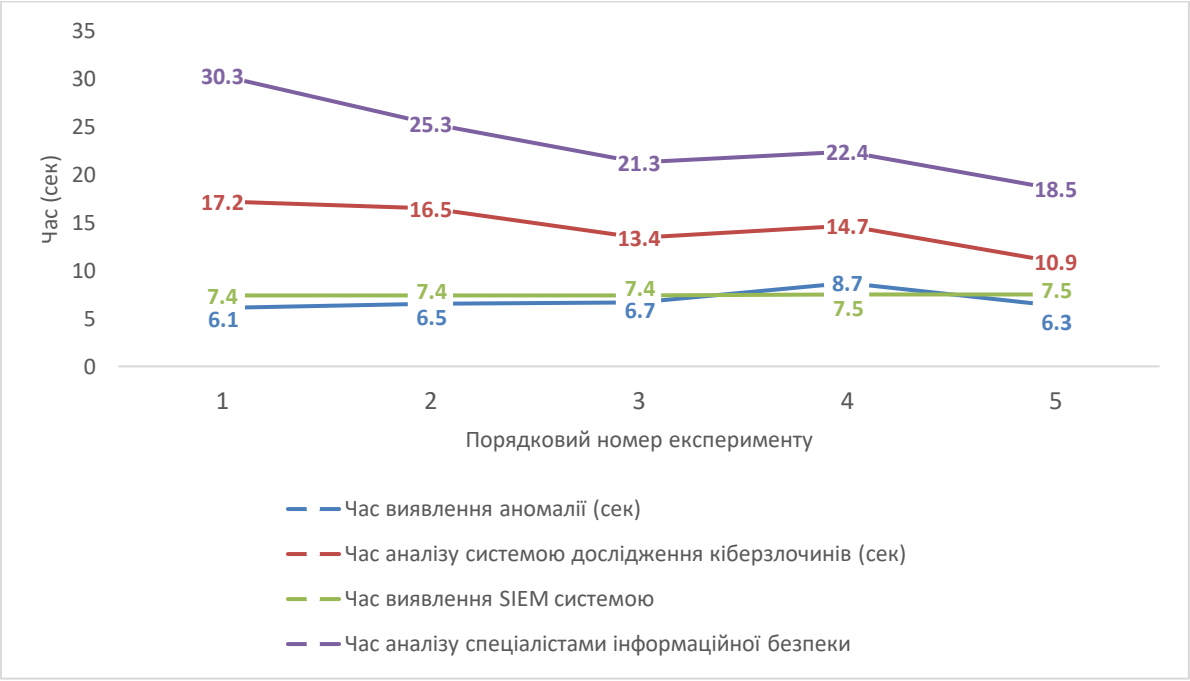



Рис. 4.9. Порівняння часу аналізу атаки Directory Traversal комплексною системою дослідження кіберзлочинів та традиційним підходом

Зважаючи на результати проведених досліджень, можна зробити висновок, що система дослідження кіберзлочинів в середньому до 7% швидше виявляє події інформаційної безпеки та може скоротити час аналізу журналів подій до 38% у відсотковому відношенні порівняно з SIEM-системою, не зменшуючи ефективність виявлення .

### 4.5.4. Аналіз швидкості виявлення та дослідження атак з порушенням логіки програм

Для вимірювання часу аналізу використовували журнали подій, згенеровані під час виконання запитів, які були виконані для порушення логіки роботи застосунку, та проведено 5 експериментів. Під час кожного експерименту використовували по 5 журналів подій, кожен з яких містив по 100 записів. Дослідження проводили протягом 5 годин по одному експерименту щогодини. Результати експерименту представлені на Рис. 4.10.

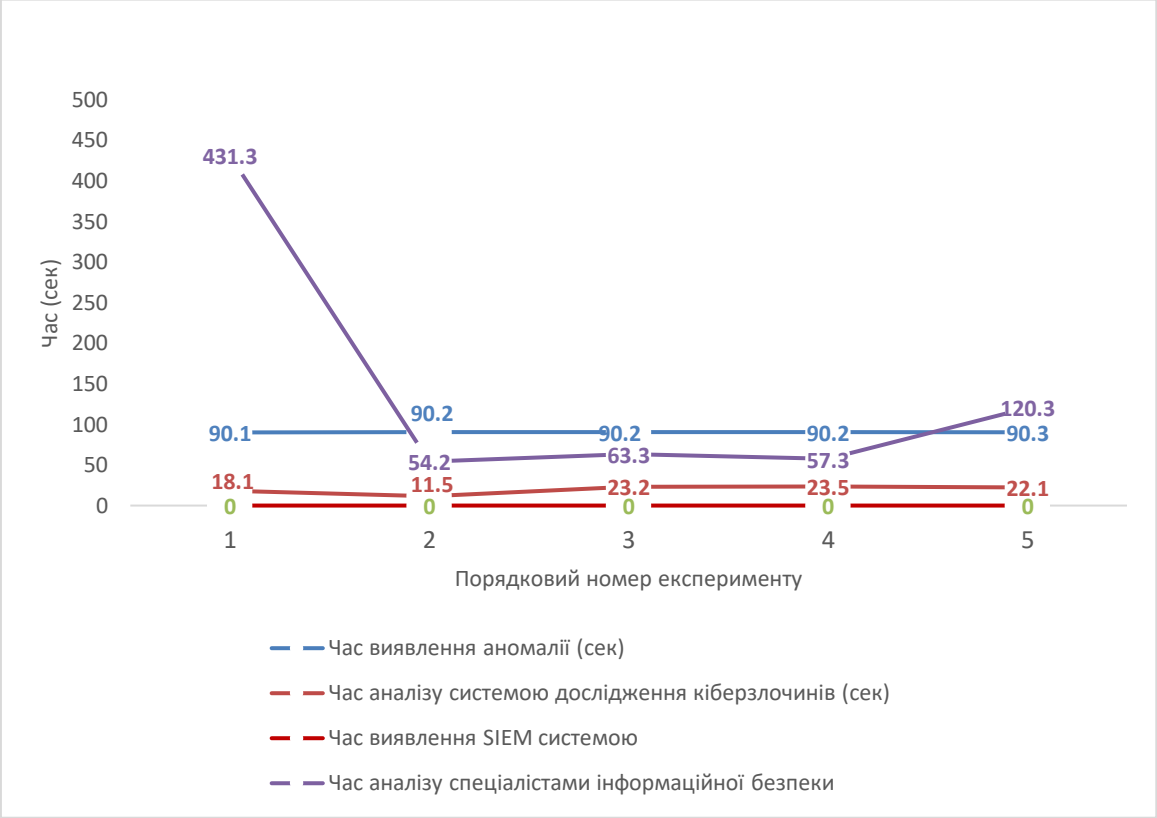



Рис. 4.10. Порівняння часу аналізу  атак з порушенням логіки комплексною системою дослідження кіберзлочинів та традиційним підходом

Середній показник часу виявлення аномалії комплексною системою дослідження кіберзлочинів становить 90.2 секунд, проте SIEM-система Wazuh з стандартним набором правил визначити даний тип атаки не змогла.

Тому наступним проведеним дослідженням став аналіз журналів подій аналітиками інформаційної безпеки та системою дослідження кіберзлочинів. Для аналітиків інформаційної безпеки в середньому необхідно було 145.28 секунд для аналізу даного типу атаки, час аналізу системою дослідження кіберзлочинів у середньому склав 19.68 секунд.

Проведений експеримент визначив, що система дослідження кіберзлочинів може в середньому скоротити час дослідження приблизно до 7 разів та виявляти запити, для яких правила виявлення розроблені попередньо не були.

### 4.5.5. Аналіз ефективності виявлення подій інформаційної безпеки системою дослідження кіберзлочинів для складових інфраструктури інформаційних систем

Експерименти, описані у попередніх підрозділах, доводять, що система дослідження кіберзлочинів для складових інфраструктури інформаційних систем загалом може швидше виявляти аномалії та швидше аналізувати кіберзлочини. Для аналізу, чи розроблена система є ефективнішою в порівнянні з SIEM-системою та чи зменшення часу впливає на відсоток виявлених атак, було проведено серію експериментів. Для цього SIEM-системі Wazuh та системі дослідження кіберзлочинів було надано для аналізу 5000 записів журналів подій, що містили 843 шкідливих записів. Результати експерименту представлено на Рис. 4.11.

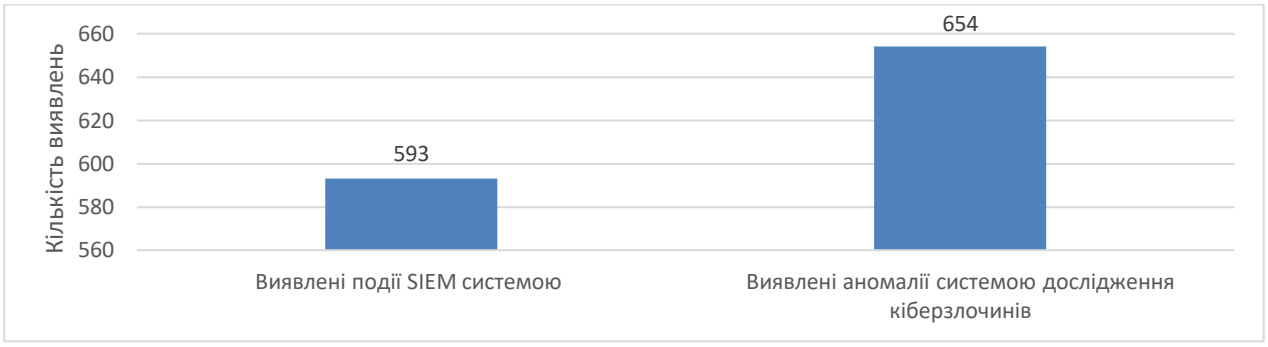



Рис. 4.11. Аналіз ефективності виявлення подій інформаційної безпеки системою дослідження кіберзлочинів для складових інфраструктури інформаційних систем

SIEM-система визначила 593 підозрілі події з проаналізованих журналів подій. 654 аномалії було виявлено системою дослідження кіберзлочинів. Обидві системи не змогли точно виявити усі шкідливі записи в журналах подій, що свідчить про потребу в подальшому тренуванні моделі ізоляційного лісу для системи дослідження кіберзлочинів. Проте, зважаючи на результати проведених експериментів, варто вказати, що система дослідження кіберзлочинів не зменшує відсоток виявлених кібератак.

### 4.6. Відповідність моделі ISO/IEC 27001:2022

Для оцінки відповідності моделі міжнародному стандарту ISO 27001:2022 було сформовано чекліст технічних контролів інформаційної безпеки на які потенційно може мати вплив модель. Процес проведення оцінки на відповідності за допомогою чекліста є структурованим та організованим підходом для перевірки рівня відповідності організації до конкретних вимог стандарту, такого як ISO 27001[101]. Впровадження комплексної системи дослідження кіберзлочинів може покращити контролі безпеки відповідно до ISO/IEC 27001:2022 та забезпечити відповідність стандарту організаціям кількома способами, зокрема:

1. Оцінка ризиків (Розділ 6). ISO 27001 підкреслює важливість управління ризиками. Запропонована система пропонує розширені можливості для виявлення та оцінки ризиків, пов’язаних з кіберзлочинністю. Це узгоджується з вимогою стандарту до організацій щодо постійної оцінки та управління ризиками інформаційної безпеки.

2. Політики інформаційної безпеки (Додаток А 5.2). Запропонована система допомагає в розробці більш надійних політик інформаційної безпеки, надаючи розуміння нових тенденцій кіберзлочинності та ефективних заходів протидії.

3. Безпека людських ресурсів (Додаток А 6.3). Система дослідження кіберзлочинів може стимулювати команди безпеки навчанню та підвищенню обізнаності, гарантуючи, що персонал ознайомлений з новітніми типами кіберзлочинності та способами реагування на них.

4. Реагування на інциденти інформаційної безпеки (Додаток А 5.26). Система дослідження кіберзлочинів може зробити значний внесок, покращивши здатність

організації виявляти, аналізувати та реагувати на інциденти безпеки своєчасно та ефективно.

5. Цикл безпечної розробки (Додаток А 8.25). Інтеграція системи дослідження кіберзлочинів, зокрема її компоненту управління вразливостями може гарантувати, що безпека є ключовим фактором у розробці та підтримці інформаційних систем.

6. Управління інцидентами інформаційної безпеки (Додаток А 5.24). Модель може покращити процеси управління інцидентами, надаючи детальну інформацію з системи дослідження загроз.

7. Аспекти інформаційної безпеки в управлінні безперервністю бізнесу (Додаток А 5.30). Система сприяє управлінню безперервністю бізнесу, допомагаючи ідентифікувати та зменшувати загрози, які можуть призвести до значних збоїв у роботі інформаційної системи.

Отже, інтеграція системи дослідження кіберзлочинів з використанням моделей ізоляційного лісу та GPT в систему управління інформаційною безпекою (ISMS) організації може покращити різні аспекти безпеки, як зазначено в ISO/IEC 27001:2022, зокрема в областях, які включають виявлення, аналіз і реагування на кіберзлочинність та інциденти безпеки.

### 4.7. Порівняльний аналіз розробленої системи з класичними SIEM

Для порівняння розробленої системи з класичними системами моніторингу подій було використано вказівки національного інституту стандартів і технологій (NIST) щодо систем SIEM. Хоча система дослідження кіберзлочинів не є SIEM-системою, проте дане порівняння дає можливість оцінити можливість самостійної роботи системи дослідження кіберзлочинів для аналізу подій інформаційної безпеки.

Стандарти NIST гарантують, що системи SIEM відповідають Федеральним стандартам обробки інформації (FIPS) та іншим відповідним актам та стандартам. Організації повинні переконатися, що інструменти SIEM відповідають цим вимогам для ефективного керування журналами та процедурам безпеки. Порівняння запропонованої системи з традиційними SIEM-системами наведено в Таблиці 4.1.

Таблиця 4.1.

Порівняння запропонованої системи з традиційними SIEM системами

| Критерій | Опис | Запропонована система | SIEM Система |
|---|---|---|---|
| Агрегація та нормалізація даних | SIEM повинні мати можливість збирати дані з різних джерел в IT-інфраструктурі організації, включаючи мережеві пристрої, сервери, програми та системи безпеки. | + (модель необхідно тренувати індивідуально для кожного типу журналів подій) | + |
| Виявлення загроз і сповіщення безпеки | SIEM повинні мати розширену аналітику для виявлення потенційних загроз безпеці. Це включає здатність аналізувати шаблони та поведінку, що вказують на зловмисну діяльність. Система повинна генерувати сповіщення про інциденти безпеки, надаючи детальну інформацію для подальшого розслідування. | + (система адаптивна до аномальної поведінки та може визначити її самостійно) | + (визначення аномальної поведінки залежить від наявності правил) |
| Відповідність міжнародним актам | Системи SIEM повинні підтримувати звітування про відповідність для різних нормативних стандартів. | + | + |

Продовження таблиці 4.1.

| | | | |
|---|---|---|---|
| Судово-медичний аналіз | Вміння проводити судово-медичний аналіз є важливим. Це включає можливості детального розслідування, щоб відстежити першопричину та наслідки інцидентів безпеки. | – | + |
| Моніторинг у режимі реального часу | Системи SIEM повинні забезпечувати можливості моніторингу в режимі реального часу для виявлення інцидентів безпеки та попередження про них. | – (конфігурується) | + |
| Зберігання даних | Необхідні відповідні можливості зберігання та утримання даних, щоб відповідати як оперативним вимогам, так і вимогам відповідності стандартам. Це включає зберігання журналів і даних подій протягом необхідного періоду. | – | + |
| Інтеграція з іншими інструментами безпеки | Системи SIEM повинні бездоганно інтегруватися з іншими інструментами безпеки для покращення безпеки.. | + | + |

Продовження таблиці 4.1.

| | | | |
|---|---|---|---|
| Аналітика поведінки користувачів і об’єктів (UEBA) | Функції UEBA для виявлення аномалій на основі поведінки користувачів і об’єктів, що допомагає ідентифікувати внутрішні загрози та скомпрометовані облікові записи. | – | + |
| Дані про загрози | SIEM має бути здатний інтегруватися із зовнішніми каналами аналізу загроз, щоб покращити свої можливості виявлення загроз за допомогою оновленої інформації про нові загрози. | + | + |
| Масштабованість і продуктивність | Система SIEM повинна бути масштабованою, щоб адаптуватися до зростаючих обсягів даних і складності IT-середовища. Продуктивність має вирішальне значення для обробки великих обсягів даних і надання своєчасної інформації. | + | + |
| Інтерфейс користувача та зручність використання | Зручний інтерфейс для ефективного моніторингу та керування сповіщеннями. | + | + |

Запропонована модель, якщо порівнювати її з класичними системами управління інформацією про безпеку та подіями (SIEM), не повністю відповідає класичним критеріям і функціям традиційних SIEM, таким як моніторинг у реальному часі та зберігання даних. Однак значно підвищує ефективність розслідування кіберзлочинів шляхом виявлення аномалій без попередньо заготовлених правил виявлення підозрілої активності. Хоча запропонована модель може не відповідати всім вимогам міжнародних стандартів до SIEM-систем, її інтеграція з існуючими засобами інформаційної безпеки є можливою. Ця інтеграція може потенційно покращити загальні заходи кібербезпеки, додавши спеціалізовані можливості розслідування кіберзлочинів. Окремий підхід і корисність моделі для детального аналізу кіберзлочинності підкреслюють її вплив на наукові дослідження в галузі кібербезпеки, вказуючи на необхідність подальших досліджень і розвитку в цій галузі. Це свідчить про те, що модель є новим додатковим інструментом до класичних систем SIEM, особливо цінним у контексті складних розслідувань кіберзлочинів.

**Висновки до розділу 4**

У четвертому розділі дисертації проведено порівняльний аналіз між запропонованою системою дослідження кіберзлочинів, що використовує моделі ізоляційного лісу та GPT і традиційними системами SIEM. Дослідження було зосереджено на різних аспектах, таких як ефективність та швидкодія дослідження кіберзлочинів. Основні висновки, зроблені в результаті проведених експериментів, включають:

1. Результати експериментів вказують, що система, реалізована на основі запропонованої моделі, демонструє швидше виявлення кіберзлочинів порівняно з традиційними системами. Зокрема у середньому для відомих типів атак на інформаційні системи (атаки типу Ін’єкції, сканування на вразливості та Directory Traversal) в середньому час виявлення зменшився до 31% та може виявляти невідомі атаки, що зумовлено здатністю навчання моделі ізоляційного лісу відрізняти нормальну поведінку від аномальної та працювати з аномаліями різного типу. Також експериментально підтверджено, що зменшення часу виявлення подій

інформаційної безпеки, запропонована система не зменшила відсоток виявлених атак.

2. Результати проведеного експерименту визначають значне зменшення часу дослідження кіберзлочинів шляхом використання моделі GPT. Зокрема у середньому під час проведених експериментів було встановлено, що час аналізу відомих атак на інформаційні системи (атаки типу Ін’єкції, сканування на вразливості та Directory Traversal) в середньому зменшився до 60%, а для аномалій, що порушують логіку роботи додатку, до 7 разів. Система дослідження кіберзлочинів значно зменшила час аналізу кіберзлочинів. Це може призвести до помітного покращення процесу дослідження кіберзлочинів.

3. Дослідження, орієнтоване на стандарт ISO 27001, демонструє, що запропонована система сприяє поліпшенню контролів у сфері управління інформаційною безпекою. Завдяки впровадженню комплексного аналізу вразливостей, інтеграції систем моніторингу загроз та детального аналізу безпекових інцидентів відбувається значне підвищення ефективності процесу управління інцидентами в області інформаційної безпеки.

Ці висновки підкреслюють переваги запропонованої системи дослідження кіберзлочинів перед традиційними системами SIEM. Здатність системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем зменшує виявлення кібератак та дослідження кіберзлочинів, не зменшуючи відсоток виявлених атак.

# ВИСНОВКИ

У даній дисертаційній роботі вирішено важливу науково-практичну проблему з підвищення ефективності виявлення кіберзлочинів в інфраструктурі інформаційних систем внаслідок використання моделей штучного інтелекту, не зменшуючи при цьому ефективність виявлення точно позитивних кібератак на різних рівнях інфраструктури інформаційної системи.

У результаті проведених експериментів було встановлено, що розроблена система дослідження кіберзлочинів ефективніша порівняно з традиційною SIEM-системою. Хоча використання запропонованої системи не може повністю замінити існуючі SIEM-системи, вона може покращити процес дослідження кіберзлочинів та пришвидшити виявлення подій інформаційної безпеки. При цьому новий підхід забезпечує зниження часу аналізу кіберзлочинів та комплексне звітування. Основні наукові і практичні результати подано нижче.

1. Аналіз основних компонентів інформаційних систем та наявних рішень виявлення вторгнень та моніторингу безпеки визначив, що наявні рішення та підходи мають обмежені можливості дослідження зовнішніх атак. Проведений аналіз відмінностей у використанні різних типів інфраструктури інформаційних систем та їх загроз визначив, що єдиної моделі системи дослідження кібербезпеки на різних рівнях інформаційної системи не існує, тому актуальною є потреба у розробці моделі системи виявлення кібератак та дослідження кіберзлочинів на різних рівнях інфраструктури інформаційних систем. Отже, підтверджено актуальність розробки нових моделей дослідження кіберзлочинів для комерційних та державних кіберсистем. Аналіз підходу DevSecOps для дослідження кіберзлочинів визначив необхідність використання сканерів вразливостей на різних рівнях інфраструктури інформаційних систем, зокрема їх інтеграцію у процес розробки, це може забезпечити своєчасну реакцію на вразливості в інформаційних системах на етапі кодування та до впровадження змін у систему.

2. Порівняльний аналіз алгоритмів штучного інтелекту на основі проведених експериментів з використанням журналів подій веб-серверу визначив, що модель з використанням ізоляційного лісу розрізняє аномалії з найнижчим рівнем хибних

спрацювань. Це підтверджено найвищим показником міри роздільності (AUC) – 0,65 під час проведених експериментів, адже чим вищий AUC, тим краще модель розрізняє позитивні та негативні класи. Ізоляційний ліс також продемонстрував найвищий показник Влучності (052), що є відношенням точно позитивних до суми хибних та точно позитивних результатів. Також, ізоляційний ліс виділився найвищим показником оцінки F1 (0,56), що є середнім гармонійним значенням влучності та відкликання. Тоді як випадковий ліс вирізнявся відкликанням, але мав проблеми з опрацюванням аномалій наданих наборів даних. Модель глибокого навчання забезпечила збалансовану продуктивність, але не найкращу точність та найвищу кількість хибних спрацювань. Зважаючи на дані проведених досліджень та аналіз роботи моделей, було прийнято рішення використовувати модель ізоляційного лісу як елементу моделі дослідження кіберзлочинів.

3. Дослідження моделей GPT встановило, що дані моделі не лише точно визначають тип кіберзлочину, але забезпечують швидке рішення для виявлення кібератак, що може мати вирішальне значення в реальних сценаріях, де час відповіді необхідно мінімізувати на основі проведених експериментів з аналізом кібератак. Зокрема, GPT-4.0 загалом демонструє підвищену ефективність обробки та виявлення різних типів кібератак порівняно з GPT-3.5. Серед усіх протестованих типів атак GPT-4.0 стабільно демонструє швидший час відповіді. Для атаки зі скануванням вразливостей і використання CVE-2021-44228 GPT-4.0 до 29.71% і 15.47% швидше аналізує журнали подій, ніж GPT-3.5 відповідно. Ці дані свідчать про те, що GPT-4.0 може бути більш ефективним при аналізі складніших типів атак. Для веб-атак, таких як обхід каталогу (як нормалізований, так і закодований) і атаки XSS, GPT-4.0 швидше до 10.94%.

4. Розроблений метод збору журналів подій з приманок на основі технології Blockchain, який завдяки децентралізованій природі Blockchain, забезпечує високу ступінь відмовостійкості та надійності журналів подій. Встановлено, що кожна транзакція або запис, який заноситься в блок, підтверджується учасниками мережі, що запобігає несанкціонованому або зловмисному редагуванню інформації.

5. Розроблена методологія дослідження кіберзлочинів з використанням системи дослідження загроз, підходу DevSecOps, моделі ізоляційний ліс та моделі GPT. Ця методологія забезпечує комплексний аналіз кіберзочинів та швидке визначення першопричини виникнення інцидентів для складових інфраструктури інформаційних систем. Дана методологія сформована з урахуванням вимог стандарту ISO/IEC 27001:2022, що забезпечує можливість її впровадження сертифікованими організаціями як для виконання вимог постійного покращення системи менеджменту інформаційної безпеки, так і для впровадження процесу управління інцидентами. Дана методологія є незалежною від типу інфраструктури інформаційної системи та від моделі GPT, що дозволяє адаптувати її під потреби організацій.

6. Розроблена модель системи дослідження кіберзлочинів, використовуючи підхід DevSecOps та моделі штучного інтелекту, з урахуванням принципів “Безпека за замовченням” та “Безпека за дизайном”, що виражено застосуванням алгоритму TLS 1.2 для передачі даних та впровадженням функціоналу маскування даних для зібраної інформації з різних систем. Функцію дослідження, що може допомогти аналітику під час опрацювання кіберзлочину виконує GPT-модель, результати з якої хоч і не повинні бути ключовими для прийняття рішення, проте мають суттєво скоротити час дослідження та визначення першопричини виникнення кіберзлочину.

7. Проведено порівняльний аналіз реакції системи дослідження кіберзлочинів на типові атаки, спрямовані на різні компоненти інфраструктури інформаційних систем, зокрема Ін’єкції, атаки з використанням сканерів на вразливості, атаки, що мають на меті порушення логіки роботи додатків та атаки типу Directory Traversal, та порівняти систему з наявними рішеннями, такими як класичні SIEM системи. Результати експериментів вказують, що для відомих типів атак на інформаційні системи (атаки типу Ін’єкції, сканування на вразливості та Directory Traversal) в середньому час виявлення зменшився до 31%, та запропонована система може виявляти невідомі атаки, що зумовлено здатністю навчання моделі ізоляційного лісу відрізняти нормальну поведінку від аномальної та працювати з аномаліями

різного типу. Експериментально підтверджено, що зменшення часу виявлення подій інформаційної безпеки, запропонована система не зменшила відсоток виявлених атак. Також система продемонструвала значне зменшення часу дослідження кіберзлочинів шляхом використання моделі GPT. У середньому під час проведених експериментів було встановлено, що час аналізу відомих атак на інформаційні системи (атаки типу Ін’єкції, сканування на вразливості та Directory Traversal) в середньому зменшився до 60% а для аномалій, що порушують логіку роботи додатка до 7 разів. Система дослідження кіберзлочинів значно зменшила час аналізу кіберзлочинів. Це може призвести до помітного покращення процесу дослідження кіберзлочинів.

8. Дослідження, зосереджене на вимогах стандарту ISO 27001:2022, виявило, що введення запропонованої системи вносить істотні удосконалення у контрольні механізми систем управління інформаційною безпекою. Встановлення комплексного аналізу вразливостей та їх усунення, інтеграція сучасних систем для відстеження загроз, а також глибокий аналіз інцидентів, пов'язаних з інформаційною безпекою, може бути впроваджене в управління інцидентами інформаційної безпеки. Встановлено, що впроваджені заходи безпеки дозволяють прогнозувати потенційні ризики, що підвищує загальний рівень захищеності інформаційних систем.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 2023 Threat Landscape Year in Review: If Everything Is Critical, Nothing Is | Qualys Security Blog. Qualys Security Blog. [Онлайн]. Доступно: https://blog.qualys.com/ vulnerabilities-threat-research /2023/12/19/2023-threat-landscape-year-in-review-part-one

2. Sobers R. 161 Cybersecurity Statistics and Trends [updated 2023]. Varonis: Automated Data Security | DSPM | AI. [Онлайн]. Доступно : https://www.varonis.com /blog/cybersecurity-statistics

3. A. A. Mughal, “Artificial Intelligence in Information Security: Exploring the Advantages, Challenges, and Future Directions”, JAMM, vol. 2, no. 1, pp. 22–34, Jan. 2018.

4. 9 Biggest Challenges of Big Data - Forbytes. [Онлайн]. Доступно: : https://forbytes.com/blog/challenges-of-big-data/

5. Serey, Joel, et al. "Pattern recognition and deep learning technologies, enablers of industry 4.0, and their role in engineering research." Symmetry 15.2 (2023): 535.

6. Phillips, Kirsty, et al. "Conceptualizing cybercrime: Definitions, typologies and taxonomies." Forensic sciences 2.2 (2022): 379-398.

7. Humayun, Mamoona, et al. "Cyber security threats and vulnerabilities: a systematic mapping study." Arabian Journal for Science and Engineering 45 (2020): 3171-3189.

8. Whyte, Christopher, and Brian Mazanec. Understanding cyber-warfare: Politics, policy and strategy. Routledge, 2023.

9. H. Kettani and P. Wainwright, "On the Top Threats to Cyber Systems," 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 2019, pp. 175-179, doi: 10.1109/INFOCT.2019.8711324.

10. Koskinen, Anna. "DevSecOps: building security into the core of DevOps." (2019).

11. "Self-Service Cybersecurity Monitoring as Enabler for DevSecOps," in IEEE Access, vol. 7, pp. 100283-100295, 2019, doi: 10.1109/ACCESS.2019.2930000.

12. Zhou, S., Liu, C., Ye, D., Zhu, T., Zhou, W., & Yu, P. S. (2022). Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. ACM Computing Surveys, 55(8), 1-39.

13. Sarker, Iqbal H. "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects." Annals of Data Science 10.6 (2023): 1473-1498.

14. Horowitz, Michael C., et al. Artificial intelligence and international security. Center for a New American Security., 2018.

15. Sakhnini, Jacob, et al. "AI and security of critical infrastructure." Handbook of Big Data Privacy (2020): 7-36.

16. Положення про технічний захист інформації в Україні, затверджене Указом Президента України від 27.09.1999 № 1229/99.

17. Порядок взаємодії органів виконавчої влади з питань захисту державних інформаційних ресурсів в інформаційних та телекомунікаційних системах, затверджений постановою Кабінету Міністрів України від 16.11.2002 р. № 1772.

18. Вимоги з ядерної та радіаційної безпеки до інформаційних та керуючих систем, важливих для безпеки атомних станцій, затверджені Наказом Державної інспекції ядерного регулювання України від 22.07.2015 № 140.

19. Mallach, E. G. (2015). Information Systems: What Every Business Student Needs to Know. Great Britain: CRC Press., pp. 38-39.

20. Проектування інформаційних систем: Загальні питання теорії проектування ІС, Конспект лекцій, Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) навч. посіб. для студ. спеціальності 122 “Комп’ютерні науки” / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. – с.88.

21. De Cesare, S., Lycett, M., Macredie, R. (2006). Development of Component-based Information Systems. Great Britain: M.E. Sharpe., p.8.

22. Boillat, Thomas; Legner, Christine (22 August 2013). "From On-Premises Software to Cloud Services: The Impact of Cloud Computing on Enterprise Software Vendors' Business Models". Journal of Theoretical and Applied Electronic Commerce Research. doi:10.4067/S0718-18762013000300004.

23. Mell, Peter, and Tim Grance. "NIST SP 800-145, The NIST definition of cloud computing." Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep (2011),

pp.1-2.

24. Lachaud, Eric. "ISO/IEC 27701 standard: Threats and opportunities for GDPR certification." Eur. Data Prot. L. Rev. 6 (2020): 194.

25. Mogull, Rich, et al. "Security guidance for critical areas of focus in cloud computing v4. 0." Cloud Security Alliance (2017), pp.8-9.

26. S. Z. Sajal, I. Jahan and K. E. Nygard, "A Survey on Cyber Security Threats and Challenges in Modern Society," 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 525-528, doi: 10.1109/EIT.2019.8833829.

27. Alouffi, Bader, et al. "A systematic literature review on cloud computing security: threats and mitigation strategies." IEEE Access 9 (2021): 57792-57807.

28. Susukailo, Vitalii, Ivan Opirsky, and Oleh Yaremko. "Methodology of ISMS Establishment Against Modern Cybersecurity Threats." Future Intent-Based Networking: On the QoS Robust and Energy Efficient Heterogeneous Software Defined Networks. Cham: Springer International Publishing, 2021. 257-271.

29. Друзюк, О. С., Волошин, Р. Я., Піскозуб, А. З., Опірський, І. Р., Сусукайло, В. А. Аналіз атак, що використовуються кіберзлочинцями під час пандемії covid 19. Захист інформації, 22(4), 220-226.

30. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the attack vectors used by threat actors during pandemic // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 261–264, 9321897, DOI: 10.1109/CSIT49958.2020.9321897 (Scopus)

31. Rehman, S., & Gautam, R. (2014). Research on access control techniques in SaaS of cloud computing. In Security in Computing and Communications: Second International Symposium, SSCC 2014, Delhi, India, September 24-27, 2014. Proceedings 2 (pp. 92-100). Springer Berlin Heidelberg.

32. Yadav, Tarun, and Arvind Mallari Rao. "Technical aspects of cyber kill chain." Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3. Springer International Publishing, 2015.

33. Isharufe, W., Jaafar, F., & Butakov, S. (2020, June). Study of security issues in platform-as-a-service (PaaS) cloud model. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) (pp. 1-6). IEEE.

34. Y. Pan, "Interactive Application Security Testing," 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 2019, pp. 558-561, doi: 10.1109/ICSGEA.2019.00131.

35. Mohan, V., Othmane, L.B.: Secdevops: is it a marketing buzzword? - mapping research on security in devops. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 542–547, August 2016

36. Susukailo, V. (2021). Використання підходу devsecops для аналізу сучасних загроз інформаційної безпеки. Електронне фахове наукове видання “Кібербезпека: освіта, наука, техніка”, 2(14), 26-35.

37. Dash, Bibhu, et al. "Threats and Opportunities with AI-based Cyber Security Intrusion Detection: A Review." International Journal of Software Engineering & Applications (IJSEA) 13.5 (2022).

38. Cao, L. (2020). AI in finance: A review. Available at SSRN 3647625.

39. Mueller, C., & Mezhuyev, V. (2022). AI Models and Methods in Automotive Manufacturing: A Systematic Literature Review. Recent Innovations in Artificial Intelligence and Smart Applications, 1-25.

40. Dharmaraj, V., & Vijayanand, C. (2018). Artificial intelligence (AI) in agriculture. International Journal of Current Microbiology and Applied Sciences, 7(12), 2122-2128.

41. Стаття 11.Кримінальний кодекс України : Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua /laws/show/2341-14#Text

42. Стаття 361-1 Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

43. Стаття 361-1. ч.1. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно:

zakon.rada.gov.ua/laws/show/2341-14#Text

44. Стаття 361-1. ч.2. Кримінальний кодекс України : Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

45. Стаття 361-2. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

46. Стаття 361-2. ч.1. Кримінальний кодекс України : Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

47. Стаття 361-2. ч.2. Кримінальний кодекс України : Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

48. Стаття 362. ч.1. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

49. Стаття 362. ч.2. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

50. Стаття 362. ч.3. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

51. Стаття 363. ч.1. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

52. Стаття 363-1. ч.1. Кримінальний кодекс України: Кодекс України від 05.04.2001 р. № 2341-III : станом на 28 берез. 2024 р. [Онлайн]. Доступно: zakon.rada.gov.ua/laws/show/2341-14#Text

53. Стаття 363-1. ч.2. Кримінальний кодекс України : Кодекс України від 05.04.2001 р. № 2341-III: станом на 28 берез. 2024 р. [Онлайн]. Доступно:

zakon.rada.gov.ua/laws/show/2341-14#Text

54. Salfati, E. and Pease, M. (2022), Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT), NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.IR.8428,

55. S Roschke, F Cheng and C Meinel, "Intrusion Detection in the Cloud", Eighth IEEE International Conference on Dependable Autonomic and Secure Computing, pp. 729-734, 2009.

56. Kostiak M., Yevseiev S., Pohasii S., Zhuchenko O., Milov O., Lysechko V., Kovalenko O., Volkov A., Lezik A., Susukailo V. Development of crypto-code constructs based on LDPC codes // Східно-Європейський журнал передових технологій. – 2022. – № 2/9 (116). – P. 44–59

57. Karen Kent, Murugiah Souppaya, Guide to Computer Security Log Management, Recommendations of the National Institute of Standards and Technology (NIST) Special Publication 800-92, September 2006.

58. Susukailo V., Opirskyy I., Kret T. Advantages of Threat Hunting with Endpoint Detection and Response Solutions // Information Protection and Security of Information Systems: VII International Scientific and Technical Conference "Information Protection and Security of Information Systems". – 2019. – Pp. 17-19.

59. Opirskyy I., Tyshyk I., Susukailo V. Evaluation of the possibility of Realizing the Crime of the Information System at Different Stages of TCP/IP // 2021 IEEE 4th International conference on advanced information and communication technologies: conference proceedings AICT-2021 (Lviv, Ukraine, September 21-25, 2021). – 2021. – С. 261–265.

60. Дослідження можливостей системи Azure Log Analytics для аналізу інцидентів інформаційної безпеки в хмарних рішеннях // Інформаційна безпека та інформаційні технології : збірник тез доповідей IV Всеукраїнської науково-практичної конференції молодих учених, студентів та курсантів (Львів, 27 листопада 2020). – 2020. – С. 57–59.

61. "From SIEM to SOC: Crossing the Cybersecurity Chasm", May 2018, RSA

Conference 2018.

62. O. Podzins and A. Romanovs, "Why SIEM is Irreplaceable in a Secure IT Environment?," 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2019, pp. 1-5, doi: 10.1109/eStream.2019.8732173.

63. S. Elder, "Vulnerability Detection is Just the Beginning," 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES, 2021, pp. 304-308, doi: 10.1109/ICSE-Companion52605.2021.00133.

64. Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

65. D. Xu, Y. Wang, Y. Meng and Z. Zhang, "An Improved Data Anomaly Detection Method Based on Isolation Forest," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2017, pp. 287-291, doi: 10.1109/ISCID.2017.202.

66. Pang, Guansong, Longbing Cao, and Charu Aggarwal. "Deep learning for anomaly detection: Challenges, methods, and opportunities." Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021.

67. Опірський І. Р., Василишин С. І., Сусукайло В. А. Аналіз загроз та безпеки технології NFC при передачі даних для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). – С. 37–44.

68. Опірський І. Р., Сусукайло В. А., Василишин С. І., Луковський Т. І. Розробка методу використання технології NFC для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). – С. 151–157.

69. Cost of a data breach 2023 | IBM. IBM in Deutschland, Österreich und der Schweiz. [Онлайн]. Доступно: https://www.ibm.com/reports/data-breach

70. Choi, Yoon-Ho, et al. "Using deep learning to solve computer security challenges: a survey." Cybersecurity 3 (2020): 1-32.

71. Chauhan, Rahul, Kamal Kumar Ghanshala, and R. C. Joshi. "Convolutional neural network (CNN) for image detection and recognition." 2018 first international conference on secure cyber computing and communication (ICSCCC). IEEE, 2018.

72. Cui, Jianjing, et al. "Comparative study of CNN and RNN for deep learning based intrusion detection system." Cloud Computing and Security: 4th International Conference, ICCCS 2018, Haikou, China, June 8-10, 2018, Revised Selected Papers, Part V 4. Springer International Publishing, 2018.

73. T. -H. Lin and J. -R. Jiang, "Anomaly Detection with Autoencoder and Random Forest," 2020 International Computer Symposium (ICS), Tainan, Taiwan, 2020, pp. 96-99, doi: 10.1109/ICS51289.2020.00028.

74. Turner, J. Rick. "Area under the curve (AUC)." Encyclopedia of Behavioral Medicine (2020): 146-146.

75. Cheung, Ronald, et al. "Diagnostic accuracy of current machine learning classifiers for age-related macular degeneration: a systematic review and meta-analysis." Eye 36.5 (2022): 994-1004.

76. Zhou, Jianlong, et al. "Evaluating the quality of machine learning explanations: A survey on methods and metrics." Electronics 10.5 (2021): 593.

77. R. G. Smith and J. Eckroth, "Building AI applications: Yesterday today and tomorrow", AI Mag., vol. 38, no. 1, pp. 6-22, 2017.

78. Sarker, Iqbal H., Md Hasan Furhad, and Raza Nowrozy. "Ai-driven cybersecurity: an overview, security intelligence modeling and research directions." SN Computer Science 2 (2021): 1-18.

79. Susukailo V., Opirskyy I., Vasilishyn S. Analysis of the possibility of using chatbots with Artificial Intelligence to detect information security incidents // Захист інформації і безпека інформаційних систем : матеріали IX Міжнародної науково-технічної конференції (Львів, 25–26 травня 2023 р.). – 2023. – С. 120–121

80. Опірський, Іван, Віталій Сусукайло, and Святослав Василишин. "Дослідження можливостей використання чатботів зі штучним інтелектом для дослідження журналів подій." Захист інформації 24.4 (2023): 177-183.

81. Flanders, Michael. "A simple and intuitive algorithm for preventing directory

traversal attacks." arXiv preprint arXiv:1908.04502 (2019).

82. Gupta, Shashank, and Brij Bhooshan Gupta. "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art." International Journal of System Assurance Engineering and Management 8 (2017): 512-530.

83. Agarwal, Yash. "Apache Log4j Logging Framework and its Vulnerability." (2022).

84. С. Василишин, І. Опірський, А. Піскозуб, Аналіз використання програмних приманок як засобу забезпечення інформаційної безпеки, DOI: 10.28925/2663-4023.2020.10.8897

85. Опірський І.Р., С.І. Василишин, В.А. Сусукайло. Розслідування кіберзлочинів за допомогою приманок у хмарному середовищі. Безпека інформації, 27(1). – с.13-20. – 2021р.. https://doi.org/10.18372/2225-5036.26.15574

86. Susukailo V., Vasilishyn S., Opirskyy I., Buriachok V., Riabchun O. Cybercrimes investigation via honeypots in cloud environments // CEUR Workshop Proceedings. – 2021. – Vol. 2923: Proceedings of selected papers of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021), Kyiv, Ukraine, January 28, 2021 (online). – p. 91–96.

87. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitaliy Susukailo, Blockchain technology as the new defense of information systems and networks in cyber-spheres, NGSEC22.

88. Vasylyshyn, S., Susukailo, V., Opirskyy, I., Kurii, Y., Tyshyk, I. (2023). A model of decoy system based on dynamic attributes for cybercrime investigation. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 6–20. doi: https://doi.org/10.15587/1729-4061.2023.273363

89. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the use of software baits (honeypots) as a means of ensuring information security // International Workshop on Information Modeling, Zbarazh, Ukraine, 2020, 2, pp. 242–245, 9321925, DOI: 10.1109/CSIT49958.2020.9321925

90. Kurii, Yevhenii, and Ivan Opirskyy. "Analysis and Comparison of the NIST SP 800-53 and ISO/IEC 27001: 2013." NIST Spec. Publ 800.53 (2022): 10.

91. Lovric, Zrinka. "Model of simplified implementation of PCI DSS by using ISO 27001 standard." Central European Conference on Information and Intelligent Systems.

Faculty of Organization and Informatics Varazdin, 2012.

92. Kitsios, Fotis, Elpiniki Chatzidimitriou, and Maria Kamariotou. "The ISO/IEC 27001 Information security management standard: how to extract value from data in the IT sector." Sustainability 15.7 (2023): 5828.

93. Lesouple, Julien, et al. "Generalized isolation forest for anomaly detection." Pattern Recognition Letters 149 (2021): 109-119.

94. Muin, Muhammad Abdul, Kapti Kapti, and Tri Yusnanto. "Campus website security vulnerability analysis using Nessus." International Journal of Computer and Information System (IJCIS) 3.2 (2022): 79-82.

95. Devi, R. Sri, and M. Mohan Kumar. "Testing for security weakness of web applications using ethical hacking." 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). IEEE, 2020.

96. Li, Jinfeng. "Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST)." arXiv preprint arXiv:2004.03216 (2020).

97. Chabchoub, Yousra, et al. "An in-depth study and improvement of Isolation Forest." IEEE Access 10 (2022): 10219-10237.

98. Pandit, Pooja D. "Nessus: Study of a Tool to Assess Network Vulnerabilities."

99. Nasereddin, Mohammed, et al. "A systematic review of detection and prevention techniques of SQL injection attacks." Information Security Journal: A Global Perspective 32.4 (2023): 252-265.

100. Asadov, Akpar. "Directory Traversal Attack." 1st INTERNATIONAL CONFERENCE ON THE 4th INDUSTRIAL REVOLUTION AND INFORMATION TECHNOLOGY. Vol. 1. No. 1. Azərbaycan Dövlət Neft və Sənaye Universiteti, 2023.

101. Опірський І. Р., Курій Є. О., Сусукайло В. А. Розробка методології оцінки відповідності стандарту ISO 27001 // Захист інформації. – 2023. – Т. 25, № 3. – С. 132–139.

# ДОДАТОК А. Акти впровадження

**ЗАТВЕРДЖУЮ**
Проректор з наукової роботи
Національного університету
«Львівська політехніка»
проф. _____ Іван ДЕМИДОВ
"___" _____ 202_ р.

**АКТ**
про використання результатів дисертаційної роботи
***Сусукайла Віталія Андрійовича***
**« Розроблення моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем»** представленої на здобуття наукового ступеня доктора філософії за спеціальністю 125 – *Кібербезпека*

Комісія у складі – голови начальника науково-дослідної частини, д.т.н., ст. досл. Небесного Р.В. та членів: завідувача кафедри захисту інформації, д.т.н, професора Опірського І.Р., завідувача відділу науково-організаційного супроводу наукових досліджень, к.т.н. Лазько Г.В. і в.о. заступника начальника планово-фінансового відділу Фаст І.І., цим актом підтверджують, що результати дисертаційної роботи Сусукайла В.А., використовувалися у відповідності до наукового напряму кафедри захисту інформації Національного університету "Львівська політехніка" - "Дослідження систем технічного захисту інформації, каналів зв'язку та комп'ютерних мереж, фізичного захисту інформації та криптографії.", в межах кафедральної науково-дослідної роботи: "Розроблення та удосконалення методів і засобів захисту інформації для протидії несанкціонованому доступу в інформаційно-комунікаційних мережах" (шифр ЗІ-7) (№ держреєстрації 0119U101690) (2019р.-2022р.);).

Сусукайлом В.А. розроблено метод збору журналів подій в інформаційних мережах з використанням систем програмних приманок, створених на основі Blockchain. Цей метод включає в себе вдосконалення математичного апарату для розрахунку динамічних атрибутів програмних приманок, який, завдяки властивостям блокчейн-технології, забезпечити цілісність журналів подій. Також, розроблено модель системи дослідження кіберзлочинів для складових інфраструктури інформаційних системи, яка використовує алгоритм Ізольованого Лісу для виявлення кібератак. Завдяки цьому підвищено швидкість виявлення кібератак .

Голова комісії,
начальник науково-дослідної
частини, д.т.н. ст. досл. _____ Роман НЕБЕСНИЙ

Члени комісії:
зав. каф. захисту інформації, д.т.н. проф. _____ Іван ОПІРСЬКИЙ

зав. відділу науково-організаційного
супроводу наукових досліджень, к.т.н. _____ Галина ЛАЗЬКО

в.о. заст. нач. планово-фінансового відділу _____ Ірина ФАСТ

**ЗАТВЕРДЖУЮ**

Проректор з науково-педагогічної роботи

Національного університету

«Львівська політехніка»

доц. _____ Олег ДАВИДЧАК

"____" _____ 202_ р.

**АКТ**

про впровадження результатів дисертаційної роботи в навчальний процес

***Сусукайла Віталія Андрійовича***

**«Розроблення моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем»** представленої на здобуття наукового ступеня доктора філософії за спеціальністю 125 – *Кібербезпека та захист інформації*

Комісія НУ «Львівська політехніка» у складі:

Голова комісії – голова науково-методичної ради інституту комп'ютерних технологій та метрології, д.т.н., проф. Байцар Р.І.

Члени комісії:

Завідувач кафедри "Захист інформації", д.т.н., проф. Опірський.І.Р, старший викладач кафедри "Захист інформації", док.філ. Василишин С.І. і доцент кафедри "Захист інформації", к.т.н., доц. Совин Я.Р. даним актом підтверджує, що проведені дисертантом наукові дослідження виконувалися ним на кафедрі «Захист інформації» Національного університету «Львівська політехніка». Основні положення та результати дисертаційної роботи впроваджені у навчальний процес кафедри «Захист інформації» Національного університету «Львівська політехніка» при вивченні дисциплін:

- «Безпека програмного забезпечення» для студентів напрямку підготовки 125 "Кібербезпека", спеціалізації «Управління інформаційною безпекою», тема №4 «OWASP Top 10» – сценарії поширених атак інфраструктурою інформаційних систем.

Голова комісії,
голова науково-методичної ради ІКТА
д.т.н., проф. Роман БАЙЦАР

Члени комісії:
зав. каф. ЗІ, д.т.н., проф. Іван ОПІРСЬКИЙ
доц. каф. ЗІ, к.т.н. доц. Ярослав СОВИН
ст.викладач каф. ЗІ, док.філ. Святослав ВАСИЛИШИН

**ЗАТВЕРДЖУЮ**
Проректор з наукової роботи
Національного університету
«Львівська політехніка»
проф. Іван ДЕМИДОВ
"02" 03 2021 р.

**АКТ**

про використання результатів дисертаційної роботи
***Сусукайла Віталія Андрійовича***
**« Розроблення моделі системи дослідження кіберзлочинів для складових інфраструктури інформаційних систем»** представленої на здобуття наукового ступеня доктора філософії за спеціальністю 125 – *Кібербезпека та захист інформації*

Комісія у складі – голови начальника науково-дослідної частини, д.т.н., ст. досл. Небесного Р.В. та членів: завідувача кафедри захисту інформації, д.т.н, професора Опірського І.Р., завідувача відділу науково-організаційного супроводу наукових досліджень, к.т.н. Лазько Г.В. і в.о. заступника начальника планово-фінансового відділу Фаст І.І., цим актом підтверджують, що результати дисертаційної роботи Сусукайла В.А., використовувалися у відповідності до наукового напряму кафедри захисту інформації Національного університету "Львівська політехніка" Окремі частини роботи виконано в межах держбюджетної науково-дослідної роботи: «Дослідження стійкості біометричних систем автентифікації до атак із застосуванням технології клонування голосу на основі глибинних нейронних мереж» (№ держреєстрації 0124U000407).

Сусукайлом В.А. розроблено метод виявлення аномалій журналів подій використовуючи алгоритм ізоляційного лісу. Цей метод включає в себе вдосконалення математичного апарату для виявлення досліджуваних атак. Завдяки цьому підвищено швидкість виявлення аномалій у журналах подій.

Голова комісії,
начальник науково-дослідної
частини, д.т.н. ст. досл. _____ Роман НЕБЕСНИЙ

Члени комісії:
зав. каф. захисту інформації, д.т.н. проф. _____ Іван ОПІРСЬКИЙ

зав. відділу науково-організаційного
супроводу наукових досліджень, к.т.н. _____ Галина ЛАЗЬКО

в.о. заст. нач. планово-фінансового відділу _____ Ірина ФАСТ

binariks

**АКТ**

**про впровадження результатів дисертаційної роботи**

**Сусукайло Віталія Андрійовича**

«РОЗРОБЛЕННЯ МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ ДЛЯ СКЛАДОВИХ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ»

Комісія у складі голови – директора Товариства з обмеженою відповідальністю «Бінарікс Україна», Шимчака Володимира Повловича, та члена комісії – експерта з кібербезпеки, Масюка Юрія-Богдана Андрійовича, склала цей акт про те, що на основі запропонованих досліджень та розглянутої моделі системи дослідження кіберзлочинів, а також математичного апарату обчислення було покращено процес виявлення подій інформаційної безпеки шляхом впровадження моделі у процес та рішення моніторингу інформаційної безпеки.

Результати дослідження, які включають розроблену модель та методологію, наразі активно використовуються для ефективного виявлення незвичайних активностей та детального аналізу інцидентів, пов'язаних з інформаційною безпекою. Ці нововведення сприяють значному покращенню процедур дослідження безпекових інцидентів, а також дозволяють компанії оперативніше реагувати на потенційні загрози.

Директор
ТОВ «Бінарікс Україна» Шимчак В.П.

Експерт з кіберезпеки
ТОВ «Бінарікс Україна» . Масюк Ю.А.

TechMagic

**А К Т**
**про впровадження результатів дисертаційної роботи**
**Сусукайло Віталія Андрійовича**

«РОЗРОБЛЕННЯ МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ ДЛЯ СКЛАДОВИХ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ»

Цей акт підтверджує використання та впровадження моделі дослідження кіберзлочинів для покращення процесів виявлення та моніторингу інцидентів інформаційної безпеки. В результаті застосування запропонованої методології, вдалося підвищити ефективність та точність визначення основної причини виникнення інцидентів інформаційної безпеки під час їх аналізу.

Також, впровадження описаної у дисертаційній роботі методології та моделі було використано, як елемент забезпечення відповідності заходам захисту визначеним стандартом ISO 27001:2022.

Директор
ТОВ «ТЕХМЕДЖИК»

# АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ

**Сусукайло Віталія Андрійовича**

«РОЗРОБЛЕННЯ МОДЕЛІ СИСТЕМИ ДОСЛІДЖЕННЯ КІБЕРЗЛОЧИНІВ ДЛЯ СКЛАДОВИХ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНИХ СИСТЕМ»

Цей акт був складений керівником Інформаційної Безпеки, як факт підтвердження результатів впровадження дисертаційної роботи. Він свідчить про те, що завдяки використанню нових досліджень, описаних у даній дисертаційній роботі, впроваджені моделі системи для аналізу кіберзлочинів та застосуванню математичних методів обчислень, у компанії «Hiveon AG» покращено методологію виявлення та моніторингу подій в сфері інформаційної безпеки та прогнозування виникнення кіберзлочинів.

В зв'язку з тим, що запропонована модель враховує сучасні тенденції в кібербезпеці та аналіз вразливостей інформаційних систем, було прийнято рішення впровадити запропоновану у роботі методологію та модель, як один з заходів захисту системи менеджменту інформаційної безпеки, що відповідає за процес реагування на інциденти інформаційної безпеки.

Керівник Інформаційної Безпеки
«Hiveon AG» Курій Є.О.

## ДОДАТОК Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

Наукові праці, в яких опубліковано наукові результати дисертації:

1. Опірський І.Р., Василишин С.І., Сусукайло В.А. Розслідування кіберзлочинів за допомогою приманок у хмарному середовищі. Безпека інформації, 27(1). – 2021. – С.13-20. *Особистий внесок здобувача: представлено порівняльну характеристику програмних приманок, проведено порівняння найпоширеніших систем дослідження кіберзлочинів у хмарах.*

2. В. Сусукайло С. Василишин, І. Опірський. Дослідження можливостей використання чатботів зі штучним інтелектом для дослідження журналів подій" // НАУ: "Захист інформації". – Том 24, №4 – Київ, 2022р. – С.177-183. *Особистий внесок здобувача: представлено порівняльну характеристику моделі GPT 3.5 та GPT 4.0;проведено експериментальне дослідження можливостей GTP моделей для аналізу експлуатації вразливості Log4j.*

3. Опірський І.Р., Василишин С.І., Сусукайло В.А. Аналіз загроз та безпеки технології NFC при передачі даних для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). С. 37-44. *Особистий внесок здобувача: проведено аналіз захищеності технології NFC.*

4. Опірський І.Р., Сусукайло В.А., Василишин С.І., Луковський Т.І. Розробка методу використання технології NFC для автоматизованої реплікації профілю користувача // Вісник Східно-Українського національного університету імені Володимира Даля. Інформаційна безпека. – 2018. – №3/4 (31/32). – С. 151–157. *Особистий внесок здобувача: проведено аналіз впливу реплікації профілю користувача на технологію NFC.*

5. Vasylyshyn, S., Susukailo, V., Opirskyy, I., Kurii, Y., Tyshyk, I. A model of decoy system based on dynamic attributes for cybercrime investigation // Eastern-European Journal of Enterprise Technologies. 2023. Vol. 1 (9 (121)). P. 6-20. (Scopus, Q3) *Особистий внесок здобувача: представлено метод збору журналів подій з*

*приманок на основі технології Blockchain, як доказів для дослідження кіберзлочинів.*

6. Сусукайло В. Використання підходу DevSecOps для аналізу сучасних загроз інформаційної безпеки // Кібербезпека: освіта, наука, техніка. – 2021. – Вип. 2, вип. 14. – С. 26–35. *Особистий внесок здобувача: представлено аналіз впливу DevSecOps підходу на ризики пов'язані з розробкою програмного забезпечення на всіх етапах SDLC; визначено Визначено набір застосунків для оптимізації процесу безпечної розробки додатків.*

7. Kostiak M., Yevseiev S., Pohasii S., Zhuchenko O., Milov O., Lysechko V., Kovalenko O., Volkov A., Lezik A., Susukailo V. Development of crypto-code constructs based on LDPC codes // Східно-Європейський журнал передових технологій. – 2022. – № 2/9 (116). – P. 44–59 *Особистий внесок здобувача: проведено аналіз кібератак на мережевому рівні інфраструктури інформаційних систем.*

8. Сусукайло В. А., Опірський І. Р., Піскозуб А. З., Волошин Р. Я., Друзюк О. С. Аналіз атак, що використовуються кіберзлочинцями під час пандемії covid 19 // Захист інформації. – 2021. – Т. 22, № 4. – С. 220–226. *Особистий внесок здобувача: представлено вектори кібератак під час пандемії COVID-19; визначено заходи захисту для протидії кібератакам під час під час пандемії COVID-19.*

9. Опірський І. Р., Курій Є. О., Сусукайло В. А. Розробка методології оцінки відповідності стандарту ISO 27001 // Захист інформації. – 2023. – Т. 25, № 3. – С. 132–139. *Особистий внесок здобувача: представлено зміни в основній частині стандарту ISO 27001:2022, що мають вплив на систему управління інформаційною безпекою.*

10. Susukailo V., Opirskyy I., Yaremko O. Methodology of ISMS Establishment Against Modern Cybersecurity Threats // Lecture Notes in Electrical Engineering. – 2022. – Vol. 831: Future intent-based networking. On the QoS robust and energy efficient heterogeneous software defined networks. – p. 257–271. *Особистий внесок здобувача: представлено методологію побудови системи управління інформаційною безпекою.*

Наукові праці, які засвідчують апробацію матеріалів дисертації:

11. Susukailo V., Opirskyy I., Kret T. Advantages of Threat Hunting with Endpoint Detection and Response Solutions // Information Protection and Security of Information Systems: VII International Scientific and Technical Conference "Information Protection and Security of Information Systems". – 2019. – Pp. 17-19. ). *Особистий внесок здобувача: представлено переваги використання технології EDR для процесу полювання на кіберзагрозу.*

12. Susukailo V. A., Opirskyy I. R. Researching the possibilities of the Azure Log Analytics system for the analysis of information security incidents in cloud solutions // Information security and information technologies: a collection of abstracts of reports of the IV All-Ukrainian scientific and practical conference of young scientists, students and cadets (Lviv, November 27, 2020). – 2020. – Pp. 57–59.). *Особистий внесок здобувача: проведено експериментальне дослідження можливостей Azure Log Analytics для аналізу подій інформаційної безпеки.*

13. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the use of software baits (honeypots) as a means of ensuring information security // International Workshop on Information Modeling. Zbarazh, Ukraine, 2020. Vol. 2, P. 242–245, 9321925. (Scopus, QX). *Особистий внесок здобувача: представлено реалізацію використання програмних приманок для дослідження подій інформаційної безпеки.*

14. Sviatoslav Vasylyshyn, Ivan Opirskyy, Vitalii Susukailo. Analysis of the attack vectors used by threat actors during pandemic // International Workshop on Information Modeling. Zbarazh, Ukraine, 2020. Vol. 2, P. 261–264, 9321897. (Scopus, QX) *Особистий внесок здобувача: представлено аналіз сучасних векторів атак зловмисників на інформаційні системи.*

15. Opirskyy I., Tyshyk I., Susukailo V. Evaluation of the possibility of Realizing the Crime of the Information System at Different Stages of TCP/IP // 2021 IEEE 4th International conference on advanced information and communication technologies: conference proceedings AICT- 2021 (Lviv, Ukraine, September 21-25, 2021). – 2021. – С. 261–265. *). Особистий внесок здобувача: представлено аналіз кібератак*

*мережевого рівня . Вдосконалено математичний апарат оцінки ймовірності реалізації мережевих атак.*

16. Susukailo V., Vasylyshyn S., Opirskyy I., Buriachok V., Riabchun O. Cybercrimes investigation via honeypots in cloud environments // CEUR Workshop Proceedings. – 2021. – Vol. 2923: Proceedings of selected papers of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021), Kyiv, Ukraine, January 28, 2021 (online). – Pp. 91–96. *Особистий внесок здобувача: представлено аналіз впливу, поведінки та ефективність програмних приманок у хмарному середовищі.*

17. Опірський І.Р., Василишин С.І. Сусукайло В.А., Дослідження вразливості Zerologon // "Технічні засоби захисту інформації", семінар при науковій раді НАН України, Київ, Україна. 2021. *Особистий внесок здобувача: представлено аналіз експлуатації вразливості Zerologon.*

18. Susukailo V., Opirskyy I., Vasylyshyn S. Analysis of the possibility of using chatbots with Artificial Intelligence to detect information security incidents // Protection of information and security of information systems: materials of the IX International Scientific and Technical Conference (Lviv, 25–26 May 2023). – 2023. – С. 120–121) *Особистий внесок здобувача: проведено експериментальне дослідження можливостей GTP моделей для аналізу ін'єкційних атак.*

# ДОДАТОК В. Фрагменти програмних кодів моделей

**Training.py**

```
# Даний скрипт використовується для перетворення журналів подій для тренування моделі Ізоляційний Ліс

import pandas as pd
import re
from urllib.parse import urlparse
from datetime import datetime
import numpy as np
def parse_log_line(line):
    log_pattern = r'(?P<ip>\d+\.\d+\.\d+\.\d+) - \[(?P<datetime>[^\]]+)\] "(?P<request>[^"]+)" (?P<status_code>\d+) (?P<response_size>\d+)'
    match = re.match(log_pattern, line)
    if not match:
        return None
    data = match.groupdict()
    method, url = data['request'].split()[:2]
    data['http_method'] = method
    data['url'] = url
    parsed_url = urlparse(url)
    data['url_length'] = len(url)
    data['path_depth'] = parsed_url.path.count('/')
    data['special_chars'] = int(any(char in url for char in ['%', '..', ';']))
    timestamp_format = '%d/%b/%Y:%H:%M:%S %z'
    data['datetime'] = datetime.strptime(data['datetime'], timestamp_format)
    return data
def read_log_file(file_path):
    with open(file_path, 'r') as file:
        log_lines = file.readlines()
    return log_lines
def preprocess_log_data(log_lines):
    parsed_lines = [parse_log_line(line) for line in log_lines if parse_log_line(line) is not None]
    return pd.DataFrame(parsed_lines)
# Прочитати дані з журналів подій
log_file_path = 'nginx.log'
log_data = read_log_file(log_file_path)

# Попередньо оброблені дані
df = preprocess_log_data(log_data)
df['hour'] = df['datetime'].dt.hour
df['day_of_week'] = df['datetime'].dt.weekday()
df.drop('datetime', axis=1, inplace=True)
```

```
df['ip_numeric'] = df['ip'].apply(lambda ip: np.sum([int(byte) for byte in ip.split('.')]))
df.drop(['ip', 'request', 'url'], axis=1, inplace=True)

# Зберегти дані в CSV
output_file = 'model_training.csv'
df.to_csv(output_file, index=False)
print(f"Дані збережені у{output_file}")
```

**IsolationForest.py**

```
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import LabelEncoder
import socket
import struct

#  Функція що конвертує IP адреси в числові значення
def ip_to_int(ip_addr):
    try:
        return struct.unpack("!I", socket.inet_aton(ip_addr))[0]
    except socket.error:
        return 0

# Функція що конвертує числові значення в IP адреси
def int_to_ip(int_ip):
    try:
        return socket.inet_ntoa(struct.pack("!I", int_ip))
    except struct.error:
        return "Invalid IP"

file_path = 'nginx.csv'  # Update this to the path of your dataset
data = pd.read_csv(file_path)

original_requests = data['request'].unique()
original_user_agents = data['user_agent'].unique()

data['remote_ip'] = data['remote_ip'].apply(ip_to_int)

request_encoder = LabelEncoder()
user_agent_encoder = LabelEncoder()

data['request'] = request_encoder.fit_transform(data['request'])
data['user_agent'] = user_agent_encoder.fit_transform(data['user_agent'])
data['datetime'] = pd.to_datetime(data['datetime'])
data['hour'] = data['datetime'].dt.hour
data['day_of_week'] = data['datetime'].dt.dayofweek
```

```
data.drop('datetime', axis=1, inplace=True)
X = data.drop('label', axis=1)

# Ініціалізувати і тренувати модель
clf = IsolationForest(random_state=42)
clf.fit(X)

# Передбачити аномадії
data['anomaly'] = clf.predict(X)

# Конвертувати закодовані дані у попередні дані
data['remote_ip'] = data['remote_ip'].apply(int_to_ip)
data['request'] = request_encoder.inverse_transform(data['request'])
data['user_agent'] = user_agent_encoder.inverse_transform(data['user_agent'])

# Зберегти дані
data.to_csv('detected anomalies.csv', index=False)
print("Виявлення аномалії завершено. Результати з оригінальними значеннями, збереженими в ‘detected anomalies.csv'.")
```

**Archery.py**

```
import requests
import hvac
from pymongo import MongoClient

# Vault налаштування
VAULT_URL = 'https://127.0.0.1:8200'
VAULT_TOKEN = 'your_vault_token'
ARCHERY_SECRET_PATH = 'secret/data/archery/credentials'
MONGODB_SECRET_PATH = 'secret/data/mongodb/credentials'

# Archery налаштування
ARCHERY_URL = 'https://localhost:8000'
LOGIN_ENDPOINT = '/api/v1/auth/login/'
REFRESH_ENDPOINT = '/v1/auth/refresh-token/'
STATIC_SCANS_ENDPOINT = '/api/v1/sast-scans/'
DYNAMIC_SCANS_ENDPOINT = '/api/v1/web-scans/'
INFRA_SCANS_ENDPOINT = '/api/v1/network-scans/'

# Ініціалізувати
client = hvac.Client(url=VAULT_URL, token=VAULT_TOKEN)

# Отримати Archery облікові дані
try:
    archery_response = client.secrets.kv.v2.read_secret_version(path=ARCHERY_SECRET_PATH)
```

```
    archery_creds = archery_response['data']['data']
    USERNAME = archery_creds['username']
    PASSWORD = archery_creds['password']
except hvac.exceptions.InvalidPath:
    print(f"Secret not found at {ARCHERY_SECRET_PATH}")
    exit(1)
except Exception as e:
    print(f"Error reading Archery credentials from Vault: {e}")
    exit(1)

# Отримати MongoDB облікові дані
try:
    mongodb_response = client.secrets.kv.v2.read_secret_version(path=MONGODB_SECRET_PATH)
    mongodb_creds = mongodb_response['data']['data']
    MONGODB_URI = mongodb_creds['uri']
    DB_NAME = mongodb_creds['db_name']
    COLLECTION_NAME = mongodb_creds['collection_name']
except hvac.exceptions.InvalidPath:
    print(f"Secret not found at {MONGODB_SECRET_PATH}")
    exit(1)
except Exception as e:
    print(f"Error reading MongoDB credentials from Vault: {e}")
    exit(1)

def get_tokens():
    """Authenticate and get access and refresh tokens."""
    response = requests.post(
        ARCHERY_URL + LOGIN_ENDPOINT,
        json={"email": USERNAME, "password": PASSWORD}
    )
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"Failed to login: {response.text}")

def refresh_access_token(refresh_token):
    """Use the refresh token to get a new access token."""
    response = requests.post(
        ARCHERY_URL + REFRESH_ENDPOINT,
        json={"refresh": refresh_token}
    )
    if response.status_code == 200:
        return response.json().get("access")
    else:
        raise Exception(f"Failed to refresh token: {response.text}")
```

```
def fetch_scan_results(access_token, endpoint):
    """Fetch scan results."""
    headers = {"Authorization": f"Bearer {access_token}"}
    response = requests.get(ARCHERY_URL + endpoint, headers=headers)
    if response.status_code == 200:
        return response.json()
    elif response.status_code == 401 and "token_not_valid" in response.text:
        return None
    else:
        print(f"Error fetching data from {endpoint}: {response.text}")
        return None

def fetch_vulnerabilities_for_scan(access_token, endpoint, scan_id):
    """Fetch vulnerabilities for a specific scan."""
    vulnerability_endpoint = f"{endpoint}{scan_id}/"
    headers = {"Authorization": f"Bearer {access_token}"}
    response = requests.get(ARCHERY_URL + vulnerability_endpoint, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching vulnerabilities for scan {scan_id}: {response.text}")
        return None

def sanitize_description(description):
    """Sanitize description to make it MongoDB compatible."""
    return description.replace('\n', ' ').replace('\r', ' ').strip()

def extract_vulnerability_data(scan_type, vulnerability):
    """Extract relevant data from a vulnerability based on scan type."""
    sanitized_description = sanitize_description(vulnerability.get('description', ''))

    if scan_type == 'Static':
        return {
            'scan_type': scan_type,
            'title': vulnerability.get('title', ''),
            'severity': vulnerability.get('severity', ''),
            'description': sanitized_description,
            'file_path': vulnerability.get('filePath', '')
        }

    if scan_type == 'Dynamic':
        return {
            'scan_type': scan_type,
            'title': vulnerability.get('title', ''),
            'severity': '',
            'description': vulnerability.get('description', ''),
```

```
            'url': vulnerability.get('url', '')
        }

    if scan_type == 'Infrastructure':
        return {
            'scan_type': scan_type,
            'title': vulnerability.get('title', ''),
            'severity': vulnerability.get('severity', ''),
            'description': sanitized_description,
            'ip': vulnerability.get('ip', '')
        }

    return {'scan_type': scan_type, 'title': 'N/A', 'severity': 'N/A', 'description': 'N/A'}

def save_vulnerabilities_to_db(scan_type, scan_data, endpoint):
    """Save vulnerabilities for each scan to MongoDB."""
    with MongoClient(MONGODB_URI) as client:
        db = client[DB_NAME]
        collection = db[COLLECTION_NAME]
        for scan in scan_data:
            scan_id = scan.get('scan_id')
            vulnerabilities = fetch_vulnerabilities_for_scan(access_token, endpoint, scan_id) or
refresh_access_token(refresh_token) and fetch_vulnerabilities_for_scan(access_token, endpoint,
scan_id)
            for vuln in vulnerabilities:
                vuln_data = extract_vulnerability_data(scan_type, vuln)
                collection.insert_one(vuln_data)

tokens = get_tokens()
access_token, refresh_token = tokens["access"], tokens["refresh"]

# Отримати результати сканування
static_vulns = fetch_scan_results(access_token, STATIC_SCANS_ENDPOINT)
dynamic_vulns = fetch_scan_results(access_token, DYNAMIC_SCANS_ENDPOINT)
infra_vulns = fetch_scan_results(access_token, INFRA_SCANS_ENDPOINT)

# Зберегти вразливості
save_vulnerabilities_to_db("Static", static_vulns, STATIC_SCANS_ENDPOINT)
save_vulnerabilities_to_db("Dynamic", dynamic_vulns, DYNAMIC_SCANS_ENDPOINT)
save_vulnerabilities_to_db("Infrastructure", infra_vulns, INFRA_SCANS_ENDPOINT)
```

**MISP.py**

```
import pandas as pd
import requests
import pymongo
from pymongo import MongoClient
import hvac
```

```
# Налаштувати з’єднання до Hashicorp Vault
vault_client = hvac.Client(url='https://your_vault_url', token='your_vault_token')
vault_data = vault_client.read('path/to/your/credentials')

# Отримати облікові дані БД та MISO
mongodb_uri = vault_data['data']['mongodb_uri']
misp_url = vault_data['data']['misp_url']
misp_key = vault_data['data']['misp_key']

# З’єднання до БД
client = MongoClient(mongodb_uri)
db = client.your_database
collection = db.ANOMALIES

# MISP headers
headers = {'Authorization': misp_key, 'Accept': 'application/json', 'Content-Type': 'application/json'}

def query_misp(value):
    response = requests.get(f"{misp_url}/attributes/restSearch", headers=headers, json={'value': value})
    if response.status_code == 200:
        data = response.json()
        # Check if the response contains any attributes
        if data.get('response') and data['response'].get('Attribute'):
            for attribute in data['response']['Attribute']:
                # Assuming 'threat_level_id' represents the associated threat level
                threat_level = attribute.get('threat_level_id', 'Unknown')
                return {'IP/Domain': value, 'Threat Level': threat_level}
        return {'IP/Domain': value, 'Threat Level': 'No threat found'}
    else:
        print(f"Error querying MISP for {value}: {response.status_code}")
        return {'IP/Domain': value, 'Threat Level': 'Error querying MISP'}

# Retrieve data from MongoDB
data = collection.find({}, {'remote_ip': 1})

results = []
for record in data:
    ip_or_domain = record['remote_ip']
    misp_result = query_misp(ip_or_domain)
    results.append(misp_result)

results_df = pd.DataFrame(results)
print(results_df)
results_df.to_csv('misp_query_results.csv', index=False)
```

**Mask.py**

```
import re

def mask_ip(input_file, output_file):
    ip_pattern = r'\b(?:\d{1,3}\.){3}\d{1,3}\b'
    mask = 'XXX.XXX.XXX.XXX'

    with open(input_file, 'r') as file:
        content = file.read()
        masked_content = re.sub(ip_pattern, mask, content)

    with open(output_file, 'w') as file:
        file.write(masked_content)

# Usage
input_log_file = 'input_log.txt'  # Replace with your input file path
output_log_file = 'masked_log.txt'  # Replace with your desired output file path

mask_ip(input_log_file, output_log_file)
```

**UI.py**

```
# Оновити та наповнити перед виконанням компонентами моделі

import tkinter as tk
from tkinter import filedialog, scrolledtext, messagebox, ttk
import subprocess
import threading
import time

def upload_data():
    file_path = filedialog.askopenfilename(filetypes=[("CSV Files", "*.csv"), ("All Files", "*.*")])
    if file_path:
        uploaded_file_label.config(text=f"Завантажений файл: {file_path}")
        with open(file_path, 'r') as file_in, open('request.txt', 'w') as file_out:
            file_out.write(file_in.read())

def simulate_progress(bar, label, message, final_message):
    for i in range(101):
        time.sleep(0.05)  # Simulate time taken for task
        bar.set(i)
        if i == 50:
            label.config(text=message)
        root.update_idletasks()
    label.config(text=final_message)

def start_analysis_thread():
    threading.Thread(target=start_analysis).start()
```

```
def start_analysis():
    try:
        subprocess.run(["python", "backend_logic.py"], check=True)
        simulate_progress(progress_var, status_label, "Відбувається аналіз...", "Аналіз завершено!")
        display_results()
    except Exception as e:
        messagebox.showerror("Error", str(e))

def aggregate_data_thread():
    threading.Thread(target=aggregate_data).start()

def aggregate_data():
    simulate_progress(progress_var, status_label, "Збираю дані...", "Збір даних завершено")

def display_results():
    try:
        with open('analysis_result.txt', 'r') as file:
            response = file.read()
            response_text.delete(1.0, tk.END)
            response_text.insert(tk.END, response)
    except Exception as e:
        messagebox.showerror("Error", str(e))

# UI Style Settings
bg_color = "#F8F9FA"
button_color = "#E9ECEF"
text_color = "#343A40"

# Set up the main application window
root = tk.Tk()
root.title("Система дослідження кіберзлочинів")
root.configure(bg=bg_color)

uploaded_file_label = tk.Label(root, text="Файл відсутній", bg=bg_color, fg=text_color)
uploaded_file_label.pack(pady=10)

upload_button = tk.Button(root, text="Завантажити дані", command=upload_data, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
upload_button.pack(pady=5)

upload_button = tk.Button(root, text="Визначити наявність аномалії", command=upload_data, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
upload_button.pack(pady=5)

aggregate_button = tk.Button(root, text="Агрегувати дані", command=aggregate_data_thread, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
```

```
aggregate_button.pack(pady=5)

start_button = tk.Button(root, text="Замаскувати дані", command=start_analysis_thread, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
start_button.pack(pady=5)

start_button = tk.Button(root, text="Демаскувати дані", command=start_analysis_thread, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
start_button.pack(pady=5)

start_button = tk.Button(root, text="Почати аналіз", command=start_analysis_thread, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
start_button.pack(pady=5)

start_button = tk.Button(root, text="Отримати результат аналізу", command=start_analysis_thread, bg=button_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
start_button.pack(pady=5)

progress_var = tk.DoubleVar()
progress_bar = ttk.Progressbar(root, orient="horizontal", length=300, variable=progress_var, mode="determinate", style="Horizontal.TProgressbar")
progress_bar.pack(pady=10)

status_label = tk.Label(root, text="Система готова до роботи", bg=bg_color, fg=text_color)
status_label.pack(pady=10)

response_text = scrolledtext.ScrolledText(root, width=70, height=20, bg=bg_color, fg=text_color, relief='flat', borderwidth=0, highlightthickness=0)
response_text.pack(pady=10)

# Configure the style of the progress bar
style = ttk.Style(root)
style.theme_use('default')
style.configure("Horizontal.TProgressbar", background=button_color, troughcolor=bg_color, bordercolor=bg_color, lightcolor=bg_color, darkcolor=bg_color)

root.mainloop()
```

**CIS.py**

```
import re
from openai import OpenAI
client = OpenAI(api_key='')
import chardet
import time

def mask_text(text):
    ip_pattern = r'\b(?:\d{1,3}\.){3}\d{1,3}\b'
```

```
    domain_pattern = r'\b(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,6}\b'
    text = re.sub(ip_pattern, 'XXX.XXX.XXX.XXX', text)
    text = re.sub(domain_pattern, 'masked.domain.com', text)
    return text

def demask_text(text, original_texts):
    # Implement logic for demasking if needed
    return text

def read_file(file_path):
    with open(file_path, 'rb') as file:
        raw_data = file.read()
        encoding = chardet.detect(raw_data)['encoding']
        return raw_data.decode(encoding)

def write_to_file(content, file_path):
    with open(file_path, 'w') as file:
        file.write(content)

def send_request_to_chatgpt(anomaly_data, flaws_data, ioc_data):

    prompt = f"Perform an analysis of the event: {anomaly_data} " \
             f"Take into account the following flaws: {flaws_data} " \
             f"and the following threats: {ioc_data}"

    attempts = 0
    max_attempts = 5
    delay = 60  # Delay in seconds

    while attempts < max_attempts:
        try:
            response = client.chat.completions.create(model="gpt-3.5-turbo",
            messages=[
                {"role": "system", "content": "You are a helpful assistant."},
                {"role": "user", "content": prompt}
            ])
            return response['choices'][0]['message']['content']
        except openai.RateLimitError:
            attempts += 1
            print(f"Rate limit exceeded, retrying in {delay} seconds...")
            time.sleep(delay)

    raise Exception("Maximum retry attempts reached.")

def main():
    anomaly_data = read_file('anomaly.csv')
```

```
    flaws_data = read_file('flaws.csv')
    ioc_data = read_file('ioc.csv')

    masked_anomaly_data = mask_text(anomaly_data)
    masked_flaws_data = mask_text(flaws_data)
    masked_ioc_data = mask_text(ioc_data)

    response = send_request_to_chatgpt(masked_anomaly_data, masked_flaws_data, masked_ioc_data)

    demasked_response = demask_text(response, [anomaly_data, flaws_data, ioc_data])

    write_to_file(demasked_response, 'analysis_result.txt')

main()
```

**PLANT UML**

**SIEM**

```
@startuml
!define RECTANGLE class

RECTANGLE SIEM {
  + Журнали подій
  + Нормалізація та аналіз
  + Механізм кореляції
  + Оповіщення та звітність
  + Аналітика безпеки
  + Реагування на інциденти
  + Розвідка загроз
  + Моніторинг активності користувачів
  + Виявлення активів
  + Виявлення загроз
  + Управління подіями
  + Визначення відповідності
  + Зберігання даних
  + Безпека мережі
  + Безпека кінцевих точок
  + Хмарна безпека
  + Інтеграція сторонніх розробників
  + Інтерфейс користувача
  + Зберігання даних
  + Операції SIEM
  + Адміністрування SIEM
  + Обслуговування SIEM
  + Моніторинг SIEM
```

```
  + Звітність SIEM
  + Інформаційна панель SIEM
}

RECTANGLE "Джерела даних безпеки" {
  + Журнали подій брандмауера
  + Журнали подій антивірусу
  + Журнали подій IDS/IPS
  + Журнали подій кінцевих точок
  + Журнали подій мережі
  + Журнали подій сервісів та додатків
}

RECTANGLE "Зовнішні джерела даних" {
  + Дані розвідки про загрози
}

SIEM -up-> "Джерела даних безпеки": Збирати журнали подій
SIEM -down-> "Зовнішні джерела даних": Отримати інформацію про загрози
@enduml
```

Система дослідження загроз

```
@startuml
!define RECTANGLE class

RECTANGLE СистемаДослідження {
  + Управління Подіями
  + Збір доказів
  + Криміналістичний аналіз
  + Розвідка загроз
  + Реагування на інциденти
  + Звітність
}

RECTANGLE ХмарнеСередовище {
  + Хмарні Сервіси
  + Віртуальні Машини
  + Контейнери
  + API запити
  + Логування
}

RECTANGLE СторонніДжерелаДаних {
  + Події безпеки
  + Дані активності користувачів
  + Сторонні джерела даних
```

```
}

СистемаДослідження -up-> ХмарнеСередовище: Зібрати Дані
СистемаДослідження -down-> СторонніДжерелаДаних: Імпортувати Інформацію
@enduml

@startuml
!define RECTANGLE class

RECTANGLE "EDR система" {
  + Агенти кінцевих точок
  + Механізм виявлення
  + Реагування на інциденти
  + Полювання на загрози
  + Звітність
}

RECTANGLE "Кінцеві точки" {
  + Кінцева Точка 1
  + Кінцева Точка 2
  + ...
  + Кінцева Точка N
}

RECTANGLE "Центральний сервер" {
  + EDR консоль
  + База даних інцидентів
  + Дані розвідки про загрозу
}

"EDR система" -up-> "Кінцеві точки": Зібрати дані
"EDR система" -down-> "Центральний сервер": Комунікація та зберігання даних
@enduml

@startuml
!define RECTANGLE class

RECTANGLE СистемаРозвідкиПроЗагрози {
  + Джерела даних про загрози
  + Збагачення даних про загрози
  + Аналіз загроз
  + Канали загроз
  + Звітність
}

RECTANGLE ЗовнішніДжерелаДаних {
```

```
  + Публічні канали загроз
  + Приватні канали загроз
  + Постачальники даних про загрози
}

СистемаРозвідкиПроЗагрози -up-> ЗовнішніДжерелаДаних: Зібрати Дані
@enduml

NIDS
@startuml
!define RECTANGLE class

RECTANGLE NIDS{
  + Ідентифікатор: int
  + Назва:string
  + Опис: string
  + Статус: string
  + Версія: string
}

RECTANGLE МережевийТрафік {
  + Вихідна IP адреса: string
  + IP призначення: string
  + Протокол: string
  + Корисне навантаження: string
}

RECTANGLE Повідомлення {
  + Ідентифікатор: int
  + Час та дата: DateTime
  + Джерело інформації: string
  + Критичність: string
  + Опис повідомлення: string
}


NIDS--|> МережевийТрафік: Аналізується
NIDS --|> Повідомлення: Генерується
@enduml

HIDS

@startuml
!define RECTANGLE class

RECTANGLE HIDS {
```

```
  + Ідентифікатор: int
  + Назва:string
  + Опис: string
  + Статус: string
  + Версія: string
}

RECTANGLE Хост {
  + Ім'я: string
  + IPАдреса: string
  + Операційна система: string
}

RECTANGLE Подія {
  + Ідентифікатор: int
  + Час та Дата: datetime
  + Джерело інформації: string
  + Тип події: string
  + Опис події: string
}

RECTANGLE Повідомлення {
  + Ідентифікатор: int
  + Час та дата: DateTime
  + Джерело інформації: string
  + Критичність: string
  + Опис повідомлення: string
}

HIDS --|> Хост: Моніторить
HIDS --|> Подія: Аналізує
HIDS --|> Повідомлення: Генерує
@enduml
```

**Атака Скануванням**

```
@startuml
actor Сканер
entity "Ціль" as Ціль

Сканер -> Сканер : ініціювати(Сканування)
Сканер -> Ціль : вибрати_ціль(Ціль)
Сканер -> Ціль : надіслати(ЗапитСканування)
Ціль -> Ціль : отримати(ЗапитСканування)
Сканер -> Сканер : проаналізуватиРезультати
```

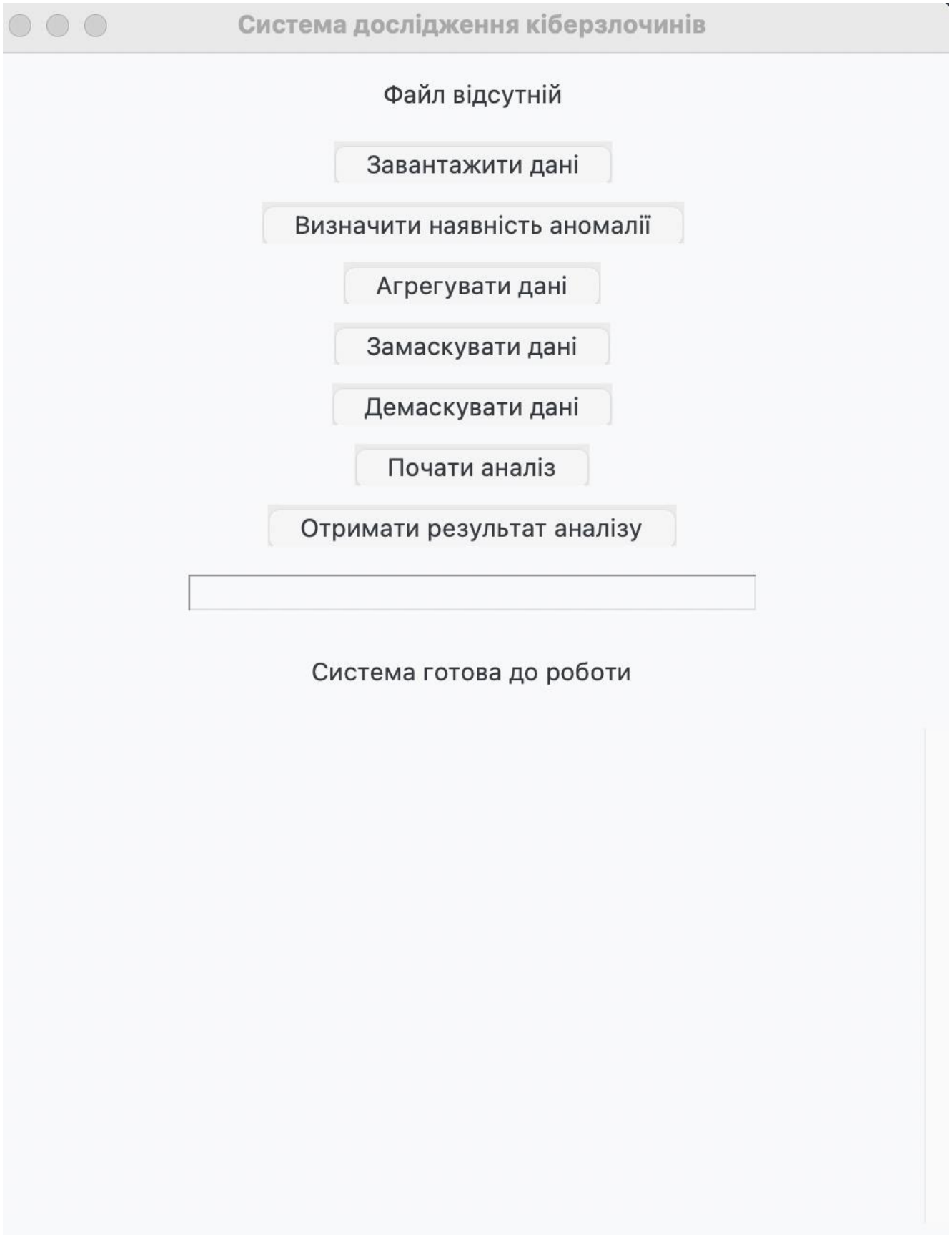
```
Сканер -> Сканер : ЗгенеруватиЗвіт(Вразливості)
@enduml
```

**Атака Ін’єкцій**

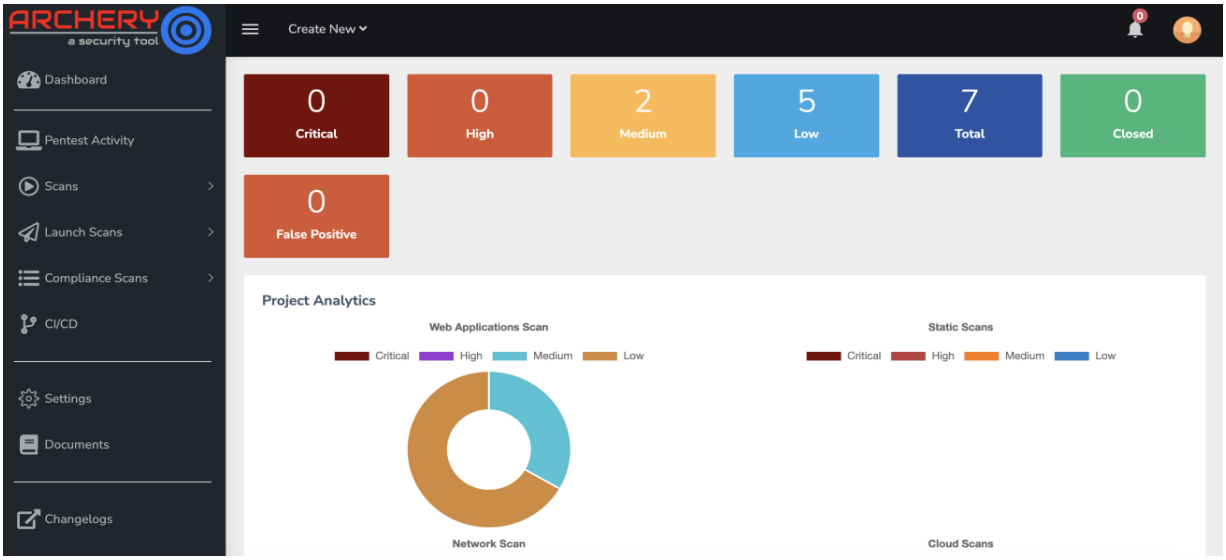
```
@startuml
actor ПрограмаТестувальник
entity "Ціль" as Ціль

ПрограмаТестувальник -> ПрограмаТестувальник : ініціювати(Ін'єкцію)
ПрограмаТестувальник -> ПрограмаТестувальник : сформувати(ШкідливийЗапит)
ПрограмаТестувальник -> Ціль : надіслати(ШкідливийЗапит, Цільt)
Ціль -> Ціль : опрацювати(ШкідливийЗапит)
Ціль -> Ціль : виконати(ШкідливийКод)
Ціль --> ПрограмаТестувальник : результат(ШкідливийРезультат)
@enduml
```

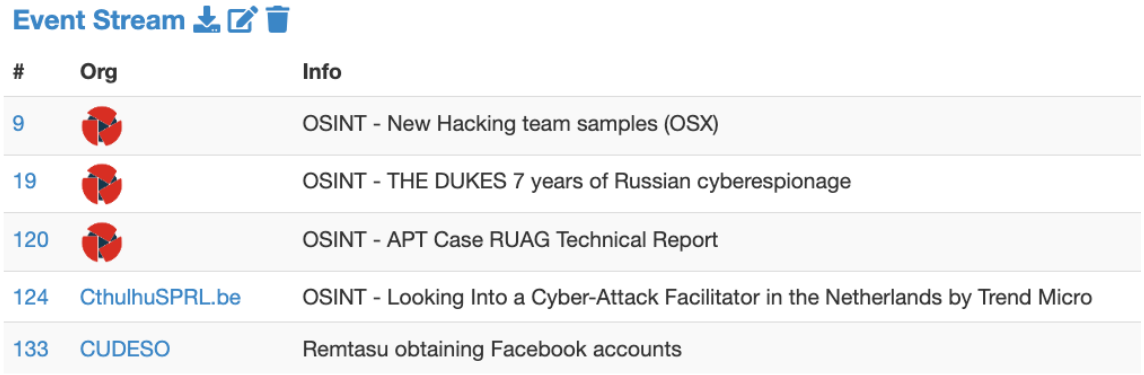
**ДОДАТОК Г. Зображення програмних компонентів запропонованої системи**




Сервіс UI




Сервіс Archety



Event Stream

| # | Org | Info |
|---|---|---|
| 9 | | OSINT - New Hacking team samples (OSX) |
| 19 | | OSINT - THE DUKES 7 years of Russian cyberespionage |
| 120 | | OSINT - APT Case RUAG Technical Report |
| 124 | CthulhuSPRL.be | OSINT - Looking Into a Cyber-Attack Facilitator in the Netherlands by Trend Micro |
| 133 | CUDESO | Remtasu obtaining Facebook accounts |

Сервіс MISP