Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«____» червня 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: Оцінка захищеності інформаційно-комунікаційної системи на основі інтелектуальних технологій

Виконавець: студентка IV курсу, групи КБ-41

_____ **Тетяна ІВАНОВА**
(підпис) (ім'я, прізвище)

	Ім'я, прізвище	**Підпис**
Керівник	Микола БРАІЛОВСЬКИЙ	

Нормоконтроль	Юрій ЩЕБЛАНІН	

Київ 2023

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студентці **КБ-41** **Івановій Тетяні Сергіївні**
(група) (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Оцінка захищеності інформаційно-комунікаційної системи на основі інтелектуальних технологій

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №3 від 20.10.2022 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Стандарти оцінки захищеності інформаційно-комунікаційних систем

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитися з затвердженими стандартами, їх розрахунками, вразливостями з боку безпеки даних, виявити можливі області для покращення, вдосконалити наявні розрахункові формул

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблений застосунок для автоматизації оцінки захищеності інформаційно-комунікаційних систем.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 24 жовтня 2022 року

Завдання видав _____ (підпис) Микола БРАІЛОВСЬКИЙ (ім'я, прізвище)

Завдання прийняла до виконання _____ (підпис) Тетяна ІВАНОВА (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	24.10.2022 – 22.01.2023	*виконано*
2	Аналіз літератури	29.02.2023 – 13.03.2023	*виконано*
3	Обґрунтування вибору рішення	13.03.2023 – 14.03.2023	*виконано*
4	Роль інтелектуальних технологій у застосунку	16.03.2023 – 03.04.2023	*виконано*
5	Аналіз проблем інформаційної безпеки в інформаційно-комунікаційних системах	05.04.2023 – 27.04.2023	*виконано*
6	Дослідження переваг та недоліків затверджених стандартів	28.04.2023 – 08.05.2023	*виконано*
7	Вдосконалення наявних розрахункових формул у стандартах та створення програмного застосунку для оцінки захищеності інформаційно-комунікаційних систем	09.05.2023 – 21.05.2023	*виконано*
8	Оформлення пояснювальної записки	21.05.2023 – 27.05.2023	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	28.05.2020 – 12.06.2023	*виконано*

Завдання видав _____ (підпис) Микола БРАІЛОВСЬКИЙ (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Тетяна ІВАНОВА (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 12 червня 2023 року

РЕФЕРАТ

Пояснювальна записка дипломної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 46 сторінки, включає в себе зміст, вступ, три розділи дипломної роботи, висновки, список джерел та 1 додаток із загальною кількістю сторінок 55. У пояснювальній записці дипломної роботи міститься 5 рисунків та 4 формули. Список використаних джерел містить 25 найменування і займає 3 сторінки.

Об'єкт дослідження — критерії оцінки захищеності інформаційно комунікаційних систем.

Мета роботи — вдосконалення наявних розрахункових формул відносно затверджених стандартів оцінки захищеності комп'ютерних систем та її автоматизація.

Результати — створений застосунок для оцінювання захищеності комп'ютерних систем від несанкціонованого доступу.

Методи дослідження:

- аналіз інформаційних ресурсів;
- порівняння існуючих підходів захищеності інформаційно-комунікаційних систем;
- моделювання методу оцінки ризиків на основі інтелектуальних технологій;
- системний підхід.

У роботі проведено порівняльний аналіз актуальних і найпоширеніших в Україні методологій ISO 27005, ДСТУ IEC/ISO 31010:2019, OCTAVE, NIST SP800-30. На їх основі розроблено комбінований підхід до оцінки ризиків на основі інтелектуальних технологій.

Програмний застосунок, розроблений в ході виконання дипломної роботи може бути використаний у інформаційно-комунікаційних системах різних за обсягом оброблювальних даних.

Напрям подальших досліджень буде спрямований на вдосконалення розробленого застосунку, візуалізації отриманих результатів, покращення рекомендацій та інтеграцію більш ширшого спектру модулів.

Ключові слова: інформаційно-комунікаційна система, кібератака, оцінка ризиків, інтелектуальні технології.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІКС	–	інформаційно-комунікаційна система
OCTAVE	–	Operationally Critical Threat, Asset, and Vulnerability Evaluation
МН	–	машинне навчання
DDoS	–	Distributed Denial of Service
CVSS	–	Common Vulnerability Scoring System
CERT	–	Computer Emergency Response Team
SOC	–	Security Operations Center

ЗМІСТ

ВСТУП

У сучасному світі глобальний бізнес зіштовхується з низкою ризиків у сфері інформаційної та кібербезпеки. Актуальність моєї роботи полягає у тому, що кіберзагрози стають все більш складними та поширеними, включаючи атаки, що використовують штучний інтелект, автоматизованих ботів та соціально-інженерні методи. Використання інтелектуальних технологій дозволяє виявляти та ліквідовувати ці загрози швидше й ефективніше.

Цей тренд пояснюється зростанням обчислювальних можливостей та обсягу збереженої інформації. Застосування аналітики даних та інтелектуальних технологій дозволяє прогнозувати майбутні кіберзагрози, розробляти стратегії захисту та виявляти слабкі місця у системах безпеки. Швидкість реагування є критичним фактором у боротьбі з кіберзагрозами. Інтелектуальні технології можуть автоматизувати процес виявлення, аналізу та реагування на загрози, що дозволяє скоротити час від реакції до реагування. Цей аналіз займає мінімум часу, що полегшує оцінку рівня безпеки власної системи та швидкість реагування на загрозу.

Оцінка захищеності інформаційно-комунікаційних систем (ІКС) включає визначення рівня вразливості і потенційних загроз, які можуть призвести до порушення конфіденційності, цілісності та доступності інформації. Головною метою цього процесу є виявлення можливих ризиків та підвищення рівня захисту ІКС.

Під час оцінки захищеності ІКС проводяться ретельний аналіз системи, її складових частин, ідентифікація потенційних вразливостей та визначення загроз, які можуть бути використані для атак або порушень безпеки. Цей процес вимагає оцінки ризиків, що можуть виникнути внаслідок вразливостей, а також оцінки важливості активів, що потребують захисту.

Оцінка захищеності ІКС допомагає виявити слабкі місця в системі та розробити план дій для зменшення ризиків. Це може включати застосування технічних, організаційних та процедурних заходів безпеки. Важливим аспектом є забезпечення відповідності системи вимогам безпеки, стандартам та регуляторним вимогам.

Оцінка захищеності ІКС є постійним процесом, оскільки загрози та вразливості можуть змінюватись з часом. Регулярне проведення оцінки дозволяє виявляти нові ризики та адаптувати заходи безпеки для ефективного захисту інформації та системи в цілому.

Метою роботи було дослідити основні проблеми, які виникають під час оцінки захищеності ІКС, розглянути ефективність традиційних методів оцінки в забезпеченні надійності захисту інформації, вдосконалити наявні розрахункові формули відносно затверджених стандартів оцінки захищеності комп'ютерних систем та автоматизувати їх. Об'єкт дослідження відповідно — критерії оцінки захищеності інформаційно комунікаційних систем.

Дослідження буде проведено покроково. Перший етап полягатиме у детальному огляді та аналізі найпоширеніших методологій оцінки захищеності інформаційно-комунікаційних систем в Україні на предмет виявлення переваг та недоліків. На другому кроці розглянемо особливості використання інтелектуальних технологій. На основі проведених досліджень будуть допрацьовані основні критерії для оцінювання ризиків. Завершальним етапом стане програмна реалізація на базі систематичного підходу.

Отримані результати можуть бути використані для подальших досліджень та впровадження в практичну діяльність організацій для забезпечення безпеки їх інформаційних систем.

РОЗДІЛ 1
АНАЛІЗ ЗАТВЕРДЖЕНИХ СТАНДАРТІВ

1.1. Критерії оцінки захищеності комп’ютерних систем

Критерії оцінки захищеності - це набір методологічних правил, що визначають певні вимоги до захисту інформації в комп'ютерних системах(КС) від несанкціонованого доступу згідно з [1]. Наразі існує велика кількість розроблених методик, що передбачають різні типи оцінки та підходи до обробки ризиків. Одні із найпоширеніших в Україні – ISO 27005, ДСТУ IEC/ISO 31010:2019, OCTAVE, NIST SP800-30.

В даній роботі були проаналізовані підходи, що досі не втратили своєї актуальності. Важливим етапом також є проведення порівняльного аналізу та вибір основи для подальшого розвитку і впровадження застосунку.

Аналіз загроз та вразливостей є важливою складовою процесу оцінки захищеності інформаційно-комунікаційних систем. Для цього проводиться ідентифікація потенційних загроз та визначення вразливостей, які можуть призвести до порушення безпеки системи.

Для аналізу загроз використовуються різні методи, за [2] включаючи експертні оцінки, статистичний аналіз і вивчення інцидентів безпеки. Експертна оцінка включає залучення фахівців з інформаційної безпеки для визначення потенційних загроз, які можуть виникнути в системі. Статистичний аналіз базується на аналізі історичних даних про загрози та використання математичних моделей для прогнозування майбутніх загроз. Вивчення інцидентів безпеки [3] дозволяє аналізувати випадки порушення безпеки, щоб виявити загрози та вразливості, які можуть бути використані проти системи.

Вразливості інформаційно-комунікаційних систем за [4] та [5] можуть бути технічними, організаційними або людськими. Технічні вразливості пов'язані з дефектами або помилками в програмному забезпеченні, налаштуваннях системи або недостатній безпеці мережі. Організаційні вразливості виникають через недостатню

політику безпеки, слабкі процедури аутентифікації або недостатню досвідченість персоналу. Людські вразливості пов'язані з соціально-інженерними атаками, фішингом або витоком інформації через недбалість або зловживання персоналу.

Для забезпечення безпеки інформаційно-комунікаційних систем важливо мати систему моніторингу та виявлення загроз. Ця система включає в себе різні інструменти та методи, які допомагають виявляти потенційні загрози та атаки.

1.1.1. ДСТУ ISO/IEC 27005:2015

ДСТУ ISO/IEC 27005:2015 [6] є національним стандартом України, який базується на міжнародному стандарті ISO/IEC 27005:2011 "Інформаційна технологія. Методи оцінки ризиків із забезпеченням безпеки". Він визначає принципи та загальні настанови для оцінки ризиків в галузі інформаційної безпеки, а також надає рекомендації та керівництво з планування, виконання та керування процесом оцінки ризиків в контексті інформаційної безпеки. Він дає методологію для виявлення та оцінки ризиків, пов'язаних з конфіденційністю, цілісністю та доступністю інформації в організації. Стандарт також враховує контекст організації, її цілі та обмеження. Він зазначає рекомендації щодо проведення оцінки ризиків, включаючи ідентифікацію активів, виявлення загроз, визначення вразливостей, оцінку впливу та вірогідності, аналіз ризиків та прийняття рішень щодо заходів з мінімізації ризиків.

Стандарт ДСТУ ISO/IEC 27005:2015 може бути використаний організаціями для покращення своєї системи управління інформаційною безпекою шляхом адекватної оцінки ризиків та прийняття ефективних заходів для забезпечення безпеки інформації.

ДСТУ IEC/ISO 31010:2019 [7] є національним стандартом України, який базується на міжнародному стандарті IEC/ISO 31010:2019 "Ризик-менеджмент - Методи оцінки ризиків". Цей стандарт надає загальні принципи, підходи та методи для оцінки ризиків в різних сферах діяльності, визначає широкий спектр методів оцінки ризиків, які можуть бути використані для управління ризиками. Він надає

організаціям рекомендації щодо вибору та застосування відповідних методів оцінки ризиків залежно від їх конкретних потреб і контексту.

Стандарт включає такі методи оцінки ризиків, як аналіз історичних даних, експертні оцінки, сценарний аналіз, аналіз впливу, кількісний аналіз, імовірнісний аналіз та багато інших. Кожен метод має свої переваги, обмеження та використовується відповідно до вимог та можливостей організації. Використання ДСТУ IEC/ISO 31010:2019 [7] допомагає організаціям зрозуміти та виявити ризики, здійснити оцінку їх впливу та імовірності, а також прийняти обґрунтовані рішення щодо управління ризиками. Цей стандарт допомагає впровадити систематичний підхід до ризик-менеджменту та забезпечує основу для розробки стратегій та заходів з мінімізації ризиків в організації.

1.1.2. Методологія OCTAVE

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) є методологією оцінки ризиків, спеціально розробленою для оцінки та управління ризиками в галузі інформаційної безпеки. OCTAVE був розроблений в рамках проекту CERT Coordination Center (CERT/CC) Карнегі-Меллонського університету.

Методологія OCTAVE дозволяє організаціям систематично оцінювати загрози, активи та вразливості, що стосуються їх інформаційних ресурсів та інфраструктури. Вона базується на розумінні контексту організації, її цілей та ризиків, і дозволяє виявити критичні активи, ідентифікувати загрози та вразливості, оцінити ризики та розробити план дій для забезпечення належного рівня захисту.

OCTAVE складається з трьох основних фаз:

1. Фаза планування: В цій фазі визначаються цілі, обсяг та контекст оцінки ризиків, формується команда, встановлюються обмеження та розробляється план роботи.

2. Фаза збору інформації: В цій фазі збирається інформація про активи, загрози та вразливості. Виконуються аналізи, інтерв'ю зі зацікавленими сторонами та оцінка ризиків.

3. Фаза аналізу та планування заходів: В цій фазі оцінюються ризики, визначаються пріоритети та розробляються плани дій для забезпечення безпеки. Результати оцінки ризиків використовуються для впровадження належних заходів з управління ризиками.

OCTAVE є ітеративним процесом, що може бути використаний для постійного вдосконалення системи управління ризиками в організації. Вона надає організаціям методологічний фреймворк для виявлення та управління ризиками інформаційної безпеки.

1.1.3. NIST SP800-30

NIST SP800-30 [8] є документом, розробленим Національним інститутом стандартів та технологій (NIST) Сполучених Штатів Америки. Він називається "Guide for Conducting Risk Assessments" і містить рекомендації та методи для проведення оцінки ризиків у сфері інформаційної безпеки.

NIST SP800-30 надає систематичний підхід до оцінки ризиків інформаційних систем. Документ визначає кроки, які слід виконати для виявлення, оцінки та керування ризиками, пов'язаними з конфіденційністю, цілісністю та доступністю інформації. У NIST SP800-30 описуються такі ключові елементи оцінки ризиків:

1. Створення контексту: Визначення мети оцінки ризиків, обсягу та обмежень, а також розуміння контексту, включаючи ідентифікацію активів, загроз та вразливостей.

2. Ідентифікація загроз: Виявлення потенційних загроз, які можуть спричинити втрату конфіденційності, цілісності або доступності інформації.

3. Визначення вразливостей: Виявлення слабких місць інформаційних систем, які можуть бути використані загрозами для виконання атак.

4. Оцінка ймовірності та впливу: Визначення імовірності виникнення загроз та оцінка можливого впливу на активи і системи.

5. Керування ризиками: Розробка стратегій та заходів для зниження ризиків до прийнятного рівня.

NIST SP800-30 є цінним документом для організацій, що працюють у сфері інформаційної безпеки, оскільки він надає методологію та рекомендації для проведення ефективних оцінок ризиків та прийняття відповідних заходів з управління ризиками.

1.2. Інтелектуальні технології захисту інформації

Один з підходів до оцінки захищеності ІКС - використання інтелектуальних технологій. Машинне навчання може бути потужним інструментом для оцінки ризиків інформаційно-комунікаційних систем. За допомогою алгоритмів машинного навчання можна побудувати моделі, які прогнозують ймовірність виникнення певних загроз або атак. Наприклад за [9] та [10], можна використовувати методи класифікації для визначення, чи належить певна активність до категорії атаки або нормальної поведінки. Моделі можуть бути навчені на основі історичних даних про загрози та їх характеристики. Такі моделі можуть допомогти виявити нові атаки або аномалії, які не були відомі раніше.

Додатково, можна використовувати методи кластеризації для групування подій та виявлення взаємозв'язків між ними. Це дозволяє ідентифікувати вразливості системи, які можуть бути використані зловмисниками для атаки.

В межах реалізації даної задачі захисту інформації у [11] досліджується зазвичай набір формальних методів, алгоритмів або моделей, які базуються на основі програмних прототипів, що дають можливість реалізувати будь-які інтелектуальні механізми захисту:

1. Автоматизоване виявлення загроз: Інтелектуальні технології, такі як машинне навчання та штучний інтелект, можуть виявляти вразливості та загрози в ІКС автоматично. Вони аналізують великі обсяги даних, виявляють аномалії та незвичайні активності, що допомагає виявити потенційні загрози швидше та ефективніше.

2. Прогнозування майбутніх загроз: Застосування аналітики даних та інтелектуальних технологій дозволяє прогнозувати майбутні кіберзагрози.

Аналізуючи дані про попередні інциденти та тренди, системи можуть прогнозувати потенційні загрози та розробляти стратегії захисту ІКС.

3. Виявлення слабких місць у системах безпеки: Інтелектуальні технології можуть допомогти виявити слабкі місця та уразливості в системах безпеки ІКС. Вони можуть аналізувати архітектуру системи, конфігураційні дані та іншу інформацію для ідентифікації можливих вразливостей.

4. Автоматизована реакція на загрози: Інтелектуальні системи можуть автоматично реагувати на виявлені загрози та інциденти в ІКС. Вони ініціюють заходи захисту, такі як блокування атак або запуск контрмір, щоб зменшити час від реакції до реагування на загрозу.

5. Аналіз ефективності захисту: Інтелектуальні технології дозволяють аналізувати ефективність застосованих заходів захисту в ІКС. Вони можуть виявляти слабкі місця, які потребують покращення, та рекомендувати вдосконалення захисних механізмів.

6. Застосування інтелектуальних технологій в оцінці захищеності ІКС допомагає автоматизувати процеси виявлення, аналізу та реагування на загрози, покращує швидкість та ефективність захисту інформації.

Машинне навчання (МН) є однією з найпопулярніших інтелектуальних технологій, яка зараз використовується для оцінки захищеності ІКС. Завдяки своїй здатності до аналізу великої кількості даних, МН може допомогти виявити вразливості та загрози, які важко виявити вручну [12].

Для використання МН у процесі оцінки захищеності ІКС необхідно мати достатньо велику базу даних для тренування моделі. Така база може включати історію атак на ІКС, характеристики захисту, збір інформації з різних джерел та інші дані.

Після тренування моделі за допомогою МН можна отримати різні показники, такі як імовірність того, що певна атака буде успішною, та рекомендації щодо поліпшення захисту ІКС. Наприклад, якщо модель показує високу імовірність того, що атака DDoS може бути успішною, то можна рекомендувати встановлення захисту від DDoS-атак.

Аналіз журналів подій за допомогою аналітики даних.

Журнали подій є джерелом важливої інформації про події, що відбуваються в ІКС. Аналіз журналів подій за допомогою аналітики даних може допомогти виявити незвичайну або підозрілу активність в ІКС, яка може бути ознакою атаки або вразливості.

Для аналізу журналів подій можна використовувати різні інструменти, такі як системи симуляції, дашборди та системи візуалізації даних. Ці інструменти допомагають виявляти залежності між подіями та візуалізувати їх для полегшення сприйняття.

Після аналізу журналів подій можна зробити висновки щодо потенційних вразливостей та загроз, які можуть призвести до порушення захисту ІКС. Наприклад, якщо аналітика даних виявляє незвичайну активність в мережі, то це може бути ознакою того, що мережу намагаються зламати.

Аналіз архітектури системи за допомогою системних аудитів

Аналіз архітектури ІКС та системних аудитів є ще одним важливим інструментом для оцінки захищеності ІКС. Ці інструменти допомагають виявляти потенційні вразливості та недоліки в архітектурі системи згідно з [13], які можуть бути використані для атаки.

Для аналізу архітектури ІКС можна використовувати різні методи, такі як аналіз потоку даних, аналіз коду та аналіз архітектурних діаграм. Ці методи допомагають виявити потенційні вразливості та недоліки в архітектурі ІКС.

Системні аудити [14] допомагають виявляти незвичайну активність та потенційні вразливості в системі. Для системних аудитів можна використовувати різні інструменти, такі як системні логи та інструменти аналізу мережевого трафіку. Ці інструменти допомагають виявити підозрілу активність та потенційні вразливості в системі.

Після аналізу архітектури ІКС та системних аудитів можна зробити висновки щодо недоліків та вразливостей в системі та рекомендувати поліпшення захисту ІКС.

Висновки за розділом 1

Загалом, інтелектуальні технології можуть значно покращити ефективність та швидкість виявлення та відповіді на кіберзагрози, допомагаючи забезпечити кібербезпеку організацій.

В роботі необхідно розробити застосунок, який повинен мати такі функціональності:

1. Введення параметрів: застосунок повинен дозволяти вводити параметри, що впливають на ризик безпеки інформаційної системи, такі як типи даних, обсяг інформації, рівень доступу, наявність заходів безпеки та інші фактори.

2. Визначення загроз: застосунок повинен мати базу даних загроз безпеці, що містить описи різних видів кіберзагроз. Він повинен дозволяти вибирати та встановлювати рівень загрози для кожного типу загрози відповідно до конкретної інформаційної системи.

3. Розрахунок ризику: На основі введених параметрів та рівнів загроз застосунок повинен провести розрахунок ризику для кожного виду загрози. Це може включати використання математичних моделей, методів аналізу та статистичних даних.

4. Візуалізація результатів: застосунок повинен відображати результати оцінки ризиків у зручному форматі, наприклад, за допомогою діаграм чи графіків. Результати повинні бути зрозумілими та легко інтерпретованими користувачами.

5. Рекомендації щодо заходів безпеки: застосунок повинен надавати рекомендації щодо необхідних заходів безпеки для зниження ризику. Це може включати пропозиції щодо вдосконалення політик безпеки, використання конкретних технологій чи методів захисту.

6. Збереження результатів: застосунок повинен дозволяти зберігати результати оцінки ризиків для подальшого аналізу та порівняння. Можливість експорту результатів у зручний формат, наприклад, файл Excel, також може бути корисною.

РОЗДІЛ 2
ВДОСКОНАЛЕННЯ СИСТЕМИ ОЦІНЮВАННЯ

2.1. Структура системи оцінювання та розрахунки

Система оцінки ризиків інформаційно-комунікаційних систем за [15] – це процес визначення, аналізу та оцінки потенційних загроз та ризиків, що стосуються безпеки та захисту ІКС. Оцінка ризиків є важливою складовою кібербезпеки і дозволяє організаціям зрозуміти, які загрози можуть вплинути на їх ІКС та які ризики вони несуть.

Common Vulnerability Scoring System (CVSS) Calculator є інструментом, який використовується для оцінки ризиків та визначення важкості вразливостей в інформаційних системах. CVSS є стандартизованою методологією, яка надає числову оцінку для кожної вразливості з метою кращого розуміння потенційних наслідків та прийняття відповідних заходів забезпечення безпеки.

CVSS оцінює вразливості на основі кількох метрик, таких як особливості, вразливості та вплив на систему. Ці метрики включають:

1. Base Metrics (Базові метрики): Вони відображають основні аспекти вразливості, такі як складність експлуатації, вплив на конфіденційність, цілісність та доступність даних. Ці метри оцінюються на основі деяких критеріїв, які надаються для кожної вразливості.

2. Temporal Metrics (Темпоральні метрики): Вони враховують часові аспекти вразливості, такі як час, що пройшов від виявлення вразливості, рівень доступності виправлення та рівень відомостей про вразливість. Ці метри враховуються при визначенні темпоральної оцінки ризику.

3. Environmental Metrics (Екологічні метрики): Вони враховують контекст конкретної інформаційної системи, на яку впливає вразливість. Ці метри включають фактори, такі як важливість системи, вартість активів, доступність контролів безпеки та інші параметри, які можуть впливати на оцінку ризику.

CVSS Calculator обчислює оцінку ризику на основі наданих даних та метрик вразливості, що вводяться. В результаті отримується числова оцінка ризику, яка допомагає організаціям визначити пріоритетність усунення вразливостей та розробку стратегій захисту. CVSS є широко використовуваним стандартом в галузі кібербезпеки і допомагає стандартизувати процес оцінки ризиків для різних вразливостей.

Використання Common Vulnerability Scoring System (CVSS) Calculator має кілька переваг:

1. Стандартизація: CVSS є широко використовуваним стандартом для оцінки ризиків вразливостей інформаційних систем. Використання CVSS Calculator дозволяє організаціям застосовувати спільний підхід до оцінки ризиків, що полегшує спілкування та порівняння результатів між різними сторонами.

2. Об'єктивність: CVSS використовує об'єктивну методологію для оцінки ризиків. Використання CVSS Calculator дозволяє систематично враховувати різні метрики вразливостей і отримувати числову оцінку ризику на основі цих метрик. Це допомагає уникнути суб'єктивних оцінок і забезпечує більш об'єктивний підхід до оцінки ризиків.

3. Порівняння ризиків: CVSS Calculator дозволяє порівнювати ризики, пов'язані з різними вразливостями. Числова оцінка ризику дозволяє зрозуміти, які вразливості мають більший потенціал негативного впливу та вимагають більш пріоритетних заходів забезпечення безпеки.

4. Планування заходів забезпечення безпеки: CVSS Calculator надає інформацію про рівень ризику для кожної вразливості. Це допомагає організаціям розробити план заходів забезпечення безпеки, пріоритезувати вразливості та визначити, на які аспекти безпеки слід звернути особливу увагу.

5. Загальна свідомість про безпеку: CVSS використовується широкою громадою кібербезпеки, що сприяє збільшенню загальної свідомості про безпеку. Використання CVSS Calculator допомагає підвищити рівень усвідомленості про ризики та сприяє впровадженню кращих практик забезпечення безпеки в організаціях.

Така реалізація має деякі недоліки, тому розглянемо кожен з пунктів більш детально:

1. Показники можливості використання

1.1. Вектор атаки. Цей показник відображає контекст, у якому можливе використання вразливостей. Цей показник прямо пропорційний віддаленості зловмисника, що може використати вразливий компонент. Він може бути охарактеризований як:

1.1.1. Мережа. Уразливість, яку можна використати за допомогою доступу до мережі. Вразливий компонент прив’язаний до мережевого стеку, а шлях зловмисника пролягає через третій рівень моделі OSI (мережевий рівень). Атака, яку можна використати як за один, так і за кілька мережевих переходів.

1.1.2. Суміжна мережа. Уразливість, яку можна використати за допомогою доступу до суміжної мережі. Вразливий компонент прив’язаний до мережевого стеку. Атака обмежена спільною фізичною або логічною мережею і її неможливо виконати через третій рівень моделі OSI (мережевий рівень).

1.1.3. Локальний. Уразливість, яку можна використати з локальним доступом. Вразливий компонент не прив’язаний до мережевого стеку, а шлях зловмисника пролягає через можливість читання/запису/виконання. Зловмисник може виконати вхід локально або взаємодіяти із вже авторизованими користувачами для запуску шкідливого файлу.

1.1.4. Фізичний. Уразливість, яку можна використати за допомогою фізичного доступу. Вимагає фізичного доступу або маніпулювання вразливим компонентом, наприклад через підключення периферійного пристрою до системи.

1.2. Складність атаки. Цей показник описує умови поза контролем зловмисника, які мають існувати для використання вразливості.

1.2.1. Низький. Спеціальних умов доступу або пом’якшувальних обставин в системі не існує. Можна очікувати повторного успіху проти вразливого компоненту.

1.2.2. Високий. Успіх атаки залежить від умов, які зловмисник не може контролювати. Успішна атака вимагає від зловмисника певної підготовки та

інвестування зусиль у збір додаткової інформації про ціль, наявність певних конфігурацій системи або обчислювальних винятків.

1.3. Привілеї. Цей показник характеризує рівень необхідних привілеїв для успішного проведення атаки.

1.3.1. Жодних. Зловмисник не має прав до атаки, тому не потребує доступу до налаштувань або на виконання файлів для використання вразливості.

1.3.2. Низький. Зловмисник має бути авторизований у системі, тобто має отримати привілеї, які надають основні можливості користувача, що зазвичай мають вплив лише на налаштування та файли, якими володіє користувач. Зловмисник має вплив лише на неконфіденційні ресурси.

1.3.3. Високий. Зловмисник має бути авторизований у системі, тобто має отримати привілеї, які надають значний контроль над уразливим компонентом, який може модифікувати параметри та файли всієї системи.

1.4. Взаємодія з користувачем. Цей показник характеризує вимоги до користувача для успішної компрометації компонента.

1.4.1. Жодного. Вразливий компонент можна використовувати без участі користувача.

1.4.2. Необхідно. Успішне використання вразливості вимагає від користувача певних дій, перш ніж уразливість можна буде використати.

1.5. Область. Цей показник характеризує здатність уразливості в одному із компонентів програмного забезпечення впливати на область поза його можливостями або привілеями.

1.5.1. Без змін. Використана вразливість може впливати лише на ресурси, якими керує той самий компонент.

1.5.2. Змінено. Використана вразливість може впливати на ресурси поза межами прав авторизації. У цьому випадку вразливий компонент і система зазнають різного впливу.

2. Показники впливу

2.1. Вплив на конфіденційність. Цей показник вимірює вплив на конфіденційність інформаційних ресурсів, якими керує програмний компонент через використану вразливість.

2.1.1. Жодного. Немає втрати конфіденційності в межах задіяного компонента.

2.1.2. Низький. Є деяка втрата конфіденційності. Отримується доступ до деякої інформації з обмеженим доступом, але зловмисник не має контролю над нею. Обмежена кількість чи вид втрат.

2.1.3. Високий. Відбувається повна втрата конфіденційності, що призводить до розкриття зловмиснику всіх ресурсів в межах вразливого компоненту. В іншому випадку можна отримати доступ до деякої конфіденційної інформації, але вона має прямий серйозний вплив на систему.

2.2. Вплив на цілісність. Цей показник вимірює вплив на цілісність інформаційних ресурсів. Цілісність означає достовірність та правдивість інформації.

2.2.1. Жодного. Немає втрати цілісності в середині пошкодженого компонента.

2.2.2. Низький. Модифікація даних можлива, але зловмисник не може контролювати наслідки або обсяг модифікації, не має прямого серйозного впливу на компонент.

2.2.3. Високий. Відбувається повна втрата цілісності або захисту. Зловмисник може змінити будь-який/всі файли.

2.3. Вплив на доступність. Цей показник вимірює вплив успішно використаної вразливості на доступність відповідного компонента.

2.3.1. Жодного. Немає жодного впливу на доступність в межах цього компоненту.

2.3.2. Низький. Є зниження продуктивності або доступності ресурсів. Ресурси в ураженому компоненті частково доступні весь час або повністю доступні лише деякий час, не має прямих серйозних наслідків.

2.3.3. Високий. Відбувається повна втрата доступності, в результаті чого зловмисник може повністю заблокувати доступ до ресурсів ураженого компонента.

Ця втрата є стійкою або постійною. Крім того, зловмисник може відмовити в певній доступності.

3. Показники темпоральної оцінки.

3.1. Зрілість коду експлойту. Цей показник вимірює ймовірність атаки на вразливість і базується на поточному стані використання методів, доступності використання коду або активності використання.

3.1.1. Не визначено. Значення не впливає на оцінку.

3.1.2. Не доведено, що експлойт існує. Код експлойту відсутній або є теоретичним.

3.1.3. Код підтвердження концепції. Доступний код експлойту для підтвердження концепції або демонстрація атаки непрактична для більшості систем. Код або техніка не функціонують у всіх ситуаціях і можуть вимагати суттєвих змін зловмисником.

3.1.4. Функціональний експлойт. Код працює в більшості ситуацій, де присутня вразливість.

3.1.5. Високий. Функціональний автономний код існує або експлойт не потрібен, а деталі широко доступні. Код експлойту працює в будь-якій ситуації або може бути доставлений автономним агентом.

3.2. Рівень виправлення. Цей показник визначає пріоритетність, коригує часовий бал у бік зменшення, відображаючи зниження терміновості остаточного рішення.

3.2.1. Не визначено. Значення не впливає на оцінку.

3.2.2. Офіційне виправлення. Доступне повне рішення постачальника, офіційний патч або оновлення.

3.2.3. Тимчасове виправлення. Є офіційне але тимчасове рішення.

3.2.4. Обхідний шлях. Існує не офіційне виправлення не від виробника. У деяких випадках користувачі самостійно створюють патч або виконують кроки, щоб обійти або зменшити вразливість.

3.2.5. Недоступний. Рішення немає або його неможливо застосувати.

3.3. Достовірність звіту. Цей показник вимірює ступінь впевненості в існуванні вразливості та достовірності відомості технічних деталей. Уразливість може бути підтверджена автором або постачальником вразливості.

3.3.1. Не визначено. Значення не впливає на оцінку.

3.3.2. Невідомо. Є повідомлення, які вказують на наявність уразливості.

3.3.3. Достовірно. Можуть бути опубліковані важливі подробиці, але експерти не мають чіткої впевненості в першоджерелі або доступу до вихідного коду, щоб повністю підтвердити взаємодію, яка призвела до отриманих результатів.

3.3.4. Підтверджено. Існують докладні звіти або можливе функціональне відтворення. Вихідний код доступний для незалежної перевірки або автор чи постачальник уразливого коду підтвердили наявність вразливості.

Незважаючи на те, що CVSS є широко використовуваним і корисним інструментом, розглянувши всі параметри та порівнявши з іншими методиками [16] можна зазначити, що він також має кілька недоліків:

1. Об'єктивність: CVSS оцінює вразливості на основі певних параметрів, які встановлюються організацією або експертами. Це може призвести до суб'єктивності в процесі оцінки, оскільки різні експерти можуть мати різні погляди на значення параметрів.

2. Складність: CVSS може бути складним і складним для розуміння. Він використовує широкий набір формул і алгоритмів для розрахунку оцінки загрози, що може створювати труднощі для неспеціалістів в області безпеки.

3. Недостатня універсальність: CVSS не завжди враховує контекст інфраструктури або конкретних застосувань. Він не враховує фактори, які можуть бути специфічними для певних ситуацій або організацій, таких як наслідки для бізнесу або можливості компрометації.

4. Відсутність динамічності: CVSS надає статичну оцінку загрози в момент часу, не враховуючи можливих змін у вразливості чи загрозах у майбутньому. Він не оновлюється автоматично, що може призвести до застарілих оцінок, коли з'являються нові відомості про вразливості.

5. Відсутність повного охоплення: CVSS орієнтований на оцінку технічних аспектів вразливостей, а не на оцінку ширшого контексту. Він не враховує соціальні, політичні або економічні фактори, які також можуть вплинути на важливість вразливостей.

Ці недоліки не означають, що CVSS неефективний, але вони свідчать про те, що важливо використовувати його в контексті і з урахуванням інших джерел інформації та експертного аналізу.

2.2. Дослідження нових метрик

Розглянутий підхід до оцінки інформаційної безпеки є ризико-орієнтованим, проте деякі метрики є недостатньо інформативними, так як передбачають бінарну відповідь, не враховують проміжні значення та суб'єктивні фактори. Впровадження нових показників та/або рівня їх впливу допоможе провести більш якісну та повну оцінку інформаційної безпеки із урахуванням кількісної складової.

Розширення критеріїв дозволяє застосовувати більш комплексний підхід до оцінки ризиків ІКС. Це означає врахування більш широкого спектру аспектів, таких як технічні, організаційні, правові, соціальні та економічні чинники, що допомагає отримати більш повне розуміння ризиків і здійснити більш об'єктивну оцінку. Розширення критеріїв оцінки ризиків ІКС дозволяє бути гнучким і адаптивним до різних контекстів та особливостей організацій. Кожна інформаційно-комунікаційна система має свої унікальні особливості, тому розширення дозволяє враховувати специфіку конкретної системи та виконувати оцінку ризиків на основі індивідуальних потреб та вимог організації. З поглибленим розумінням ризиків та їх оцінкою з різних перспектив, організації можуть розробляти та впроваджувати більш ефективні заходи забезпечення безпеки для своїх ІКС.

Першим кроком у розробці методики оцінки захищеності інформаційно-комунікаційних систем на основі інтелектуальних технологій є визначення цілей оцінки. Цілі згідно з [17] можуть включати:

1. Визначення загального рівня захищеності ІКС: Оцінка загального стану безпеки системи та визначення рівня вразливостей та загроз.

2. Виявлення слабких місць: Виявлення конкретних вразливостей та ризиків, що можуть бути використані зловмисниками.

3. Оцінка ефективності заходів захисту: Оцінка того, наскільки ефективно існуючі заходи захисту здатні запобігти загрозам та атакам.

4. Прогнозування майбутніх загроз: Використання інтелектуальних технологій для прогнозування майбутніх загроз та визначення потенційних сценаріїв атак.

Після визначення цілей оцінки захищеності ІКС потрібно вибрати відповідні інтелектуальні технології [10], які допоможуть досягти цих цілей. Деякі з популярних інтелектуальних технологій, які можуть бути використані для оцінки захищеності ІКС, включають:

1. Методи машинного навчання: Використання алгоритмів навчання на основі даних для виявлення аномальної активності, класифікації загроз, аналізу вразливостей та прогнозування майбутніх атак.

2. Аналітика великих даних: Використання методів аналізу великих обсягів даних для виявлення складних залежностей та патернів, які можуть свідчити про потенційні загрози.

3. Аналіз поведінки користувачів: Використання методів аналізу поведінки користувачів для виявлення відхилень від типового поведінкового шаблону, що може свідчити про зловмисну активність.

4. Моделювання загроз: Використання методів моделювання для відтворення сценаріїв атак та оцінки їх впливу на ІКС.

Для проведення об'єктивної оцінки захищеності ІКС необхідно розробити метрики та критерії відносно [18] та [19], які дозволять виміряти рівень безпеки та оцінити ефективність заходів захисту.

Досліджені у першому розділі стандарти дозволяють скорегувати кількісну оцінку. Спираючись на [20], [21] розглянемо перший блок критерій «Показники можливості використання». Метрика, що відповідає за визначення складності атаки, передбачає лише два варіанти: високий і середній. В той самий час додатком

вищезазначеного стандарту ідентифікується п’ять рівнів (дуже високий, високий, середній, низький та дуже низький).

Схожа проблема супроводжує критерій визначення привілеїв – не вистачає позицій, які б більш чітко та широко характеризували рівень необхідних привілеїв. Тому в своєму додатку впроваджую наступні рівні оцінки: Жодних, Користувач системи, Адміністратор, Власник системи.

Наступним кроком розглянемо показники впливу. В кожному із трьох не передбачений усереднений рівень впливу. У запропонованому застосунку визначені три рівні впливу: жодного, низький, високий. Таке розмежування не дає в повній мірі якісно оцінити рівень впливу, що в подальшому залишає за собою можливість спекулювання фінальними результатами. Темпоральні метрики є репрезентативними і повною мірою відображають вплив, пов’язаний із використанням експлойту. Тому дані параметри залишені без змін.

Звітність є важливою складовою даного розділу роботи за [22], оскільки вона дозволяє систематизувати і представити результати оцінки захищеності ІКС на основі інтелектуальних технологій. Звіти повинні бути структурованими, докладними та зрозумілими, щоб забезпечити доступність інформації для зацікавлених сторін. Вони повинні містити наступні розділи:

1. Результати оцінки: Цей розділ містить основні результати оцінки захищеності ІКС [23]. Він може включати виявлені загрози, вразливості, оцінку ризиків та ефективність заходів захисту. Результати можуть бути представлені у вигляді таблиць, графіків та інших візуалізацій, що полегшують їх сприйняття.

2. Рекомендації: У цьому розділі формуються рекомендації щодо поліпшення захищеності ІКС на основі виявлених загроз та вразливостей [24]. Рекомендації можуть стосуватися вдосконалення існуючих заходів захисту, впровадження нових технологій чи політик безпеки.

3. Висновки: В цьому розділі наводяться загальні висновки з оцінки захищеності ІКС на основі інтелектуальних технологій [25]. Підкреслюються головні недоліки, переваги та можливі напрямки подальшого розвитку і покращення системи.

2.3. Впровадження метрик у розрахункові формули

Формула розрахунку CVSS згідно із [8] складається з трьох основних компонентів: базових метрик (Base Metrics), середніх метрик (Temporal Metrics) і серйозних метрик (Environmental Metrics). Кожна компонента використовує свої власні підметрики для визначення числових значень. Основна формула для розрахунку CVSS виглядає наступним чином:

$$CVSS = (AV \times AC \times PR \times UI \times S) + (C \times I \times A), \quad (2.1)$$

де AV - базовий вектор атаки (Attack Vector)

AC - складність атаки (Attack Complexity)

PR - наявність аутентифікації (Privileges Required)

UI - вплив використання аутентифікації (User Interaction)

S - базова серйозність (Scope)

C - вплив на конфіденційність (Confidentiality)

I - вплив на цілісність (Integrity)

A - вплив на доступність (Availability)

Кожна з цих підметрик має свої можливі значення, які можуть бути числовими або категоріальними. Ці значення використовуються для розрахунку числового значення CVSS. Зазвичай, кожна підметрика має шкалу від 0 до 10, де 0 - найнижчий рівень ризику, а 10 - найвищий рівень ризику. Рівняння кожного основного компонента визначені нижче із урахуванням нововведень.

2.3.1. Базовий показник

Базовий показник є функцією рівнянь підрахунків впливу та можливості використання. Де базова оцінка визначається як,

If(Impact sub score <= 0) 0 else,

$$ISC_{\text{Base}} = RoundUp\left(min\left(\left(1-(1-AV)\times(1-AC)\times(1-PR)\times (1-UI)\right)\times\left(\left(S\times(C+I+A)\right)-0.029\right)-3.25\right),1\right), \quad (2.2)$$

де AV - базовий вектор атаки (Attack Vector)

AC - складність атаки (Attack Complexity)

PR - наявність аутентифікації (Privileges Required)

UI - вплив використання аутентифікації (User Interaction)

S - базова серйозність (Scope)

C - вплив на конфіденційність (Confidentiality)

I - вплив на цілісність (Integrity)

A - вплив на доступність (Availability)

0.029 – у формулі відповідає константному значенню, яке використовується для нормалізації і збалансування впливу коефіцієнта вразливості в рамках формули базових метрик

3.25 – у формулі відповідає константному значенню, яке використовується для зміщення (offset) остаточного результату розрахунку; значення вибрано таким чином, щоб додати до остаточного результату відповідну вагу та вплив на оцінку загальної серйозності вразливості.

І допоміжний показник експлуатаційної здатності,

$$8{,}22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction\ , \quad (2.3)$$

де AV - базовий вектор атаки (Attack Vector)

AC - складність атаки (Attack Complexity)

PR - наявність аутентифікації (Privileges Required)

2.3.2. Темпоральні показники

Темпоральний бал визначається як,

$$AdjustedTemporal = RoundUp(BaseScore \times ExploitCodeMaturity \times RemediationLevel \times ReportConfidence, 1)\ , \quad (2.4)$$

де AdjustedTemporal - адаптована темпоральна метрика (Adjusted Temporal)

BaseScore - базова метрика (Base Score)

ExploitCodeMaturity - ступінь готовності експлойта (Exploit Code Maturity)

RemediationLevel - рівень усунення (Remediation Level)

ReportConfidence - рівень довіри до звіту (Report Confidence)

Де «Round up» визначається як найменше число з точністю до одного знака після коми, яке дорівнює або перевищує введене значення. Наприклад, Round up (4.02) дорівнює 4.1; а Round up (4,00) дорівнює 4,0.

Висновки за розділом 2

У процесі дослідження наявних метрик був зроблений висновок щодо нестачі певного проміжного рівня впливу. Тому задля отримання більш точного результату було введено поняття середнього рівня впливу показника. На основі даних метрик та розрахункових формул було прийнято рішення покращити наявний застосунок та реалізувати його у вигляді програмного застосунку для операційної системи Windows. Відповідні зміни будуть додані в розрахунки. Завершуючи другий розділ, була розроблена методика оцінки захищеності інформаційно-комунікаційних систем на основі інтелектуальних технологій. Ця методика включає визначення цілей оцінки, вибір інтелектуальних технологій, розробку метрик та критеріїв оцінки, розробку процедур оцінки та інструментів та документування результатів оцінки. Дана методика може бути використана для оцінки захищеності ІКС з метою виявлення загроз, вразливостей та поліпшення ефективності заходів захисту.

РОЗДІЛ 3
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОЦІНКИ ЗАХИЩЕННОСТІ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1. Програмна реалізація

Задача полягає в створенні програмного калькулятора для розрахунку CVSS з використанням Windows Forms і мови програмування C#.

Ця програма є калькулятором, що використовується для оцінки захищеності інформаційно-комунікаційних систем. Вона надає можливість ввести значення різних метрик, необхідних для розрахунку оцінки загрози, і повертає результати базової та темпоральної метрик.

Інтерфейс програми реалізований за допомогою Windows Forms, що дозволяє користувачеві обрати необхідну оцінку кожної метрики для визначення значень коефіцієнтів у одному із одинадцяти listBox компонентів та отримати результатів розрахунків. Всі введені дані перетворюються в числовий формат для подальшого використання. Для зручності метрики відокремлені на різних вкладках. Переключення вкладок можливе як за натиском кнопки «Далі», так і безпосередньо за самими вкладками. Навівши курсор на кожну метрику користувач отримає спливаючу підказку із визначенням та характеристикою відповідної.

Програма має наступні функціональні можливості:

1. Введення значень коефіцієнтів:

- Вектор атаки:
- Складність атаки
- Необхідні привілеї
- Взаємодія з користувачем
- Область впливу
- Вплив на конфіденційність
- Вплив на цілісність

- Вплив на доступність
- Зрілість коду експлойту: рівень зрілості експлойту або інструменту
- Рівень виправлення: рівень усунення вразливості
- Достовірність звіту: рівень довіри до звіту про вразливість

2. Розрахунок базової метрики:

- Для визначення базової метрики необхідно заповнити всі дані на вкладках «Використання» та «Вплив», вони є обов’язковими.

3. Розрахунок темпоральної метрики:

- Темпоральні метрики враховують чинники, що змінюються з часом, такі як рівень зрілості експлойту, рівень усунення вразливості та рівень довіри до звіту.

4. Відображення результатів розрахунків:

- Результати базової та темпоральної метрик відображаються на вкладці результатів у вигляді діаграм.

5. Збереження результатів:

- Таблиці із даними можна зберегти на пристрій натиснувши кнопку «Зберегти».

Усі ці етапи програми відображені на рис. 3.1. Для візуалізації обраних позицій користувач може натиснути на кнопку «Оновити». У обробнику події кнопок "Оновити" (button3_Click, button4_Click) отримуються значення коефіцієнтів з полів, а потім заповнюється таблиця за допомогою відповідних функцій. Тоді дані з’являться на екрані у вигляді таблиці, що передбачає вивід метрики і рівня її впливу.

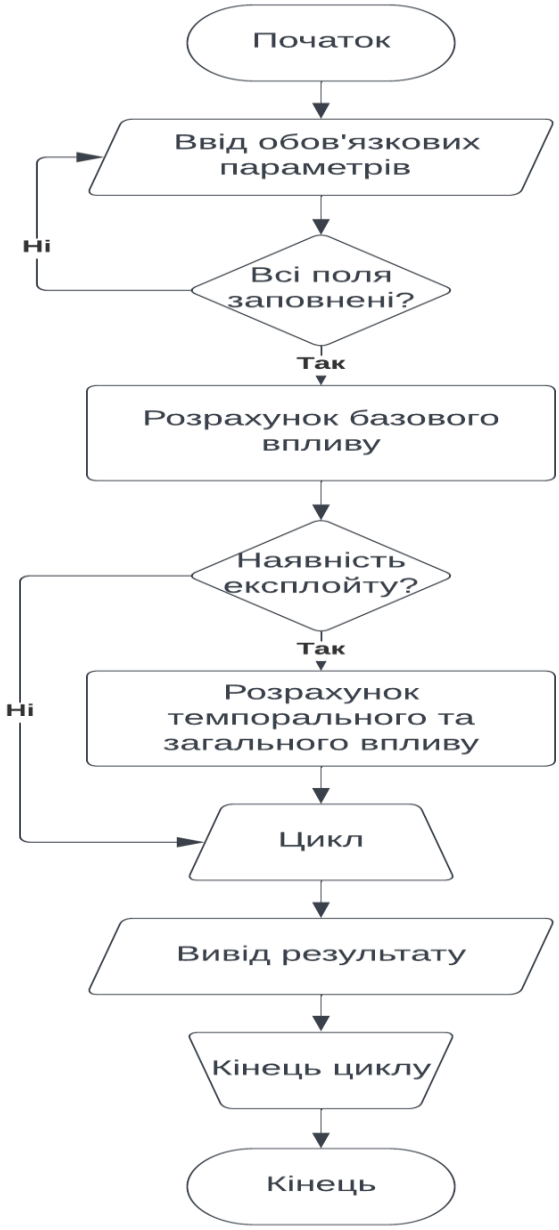

Рисунок 3.1 – Життєвий цикл програми

Після заповнення необхідних показників користувачеві необхідно натиснути на кнопку "Розрахувати" для отримання результатів розрахунків. У обробнику події кнопки "Розрахувати" (button6_Clic) отримуються значення коефіцієнтів з текстових полів, а потім виконується перевірка заповненості обов’язкових полів, автоматичний перевід на відповідну вкладку та вивід фінального результату у вигляді діаграми. У випадку відсутності одного із коефіцієнтів, користувач отримає відповідне повідомлення у вигляді позначки помилки біля кнопки розрахунку.

Для збереження результатів у форматі таблиці Excel були використані сторонні бібліотеки, такі як EPPlus та NPOI, які надають можливість створювати та заповнювати файли Excel у програмі на C#.

3.2. Тестування

Тестування програми є необхідною складовою розробки програмного забезпечення для виявлення помилок та дефектів у програмі. Навіть дрібні помилки можуть призвести до неправильної роботи програми, що може вплинути на її функціональність та надійність.

Тестування допомагає забезпечити високу якість програмного забезпечення. Це включає перевірку відповідності програми вимогам, правильність розрахунків, коректність введення та виведення даних, а також перевірку на відповідність стандартам та критеріям якості. Добре протестована програма надає впевненість користувачам у її працездатності та надійності.

При використанні компонента ListBox для введення даних (рис. 3.2) можуть виникати деякі типові помилки:

1. Відсутність вибору: Якщо ви вимагаєте вибір з ListBox, слід перевірити, що користувач обрав потрібний варіант перед продовженням. Якщо користувач не вибрав жодного варіанту, потрібно відобразити повідомлення про помилку або встановити значення за замовчуванням.

2. Проблеми з розміщенням: При великій кількості елементів в ListBox може виникнути проблема зі зручністю прокручування та вибору елементів.

3. Перевантаження інтерфейсу: З великою кількістю ListBox або багатьма компонентами на формі може стати важко використовувати програму. Потрібно врахувати ергономіку та зручність користування, використовуючи відповідну організацію елементів і можливість групувати схожі дані.

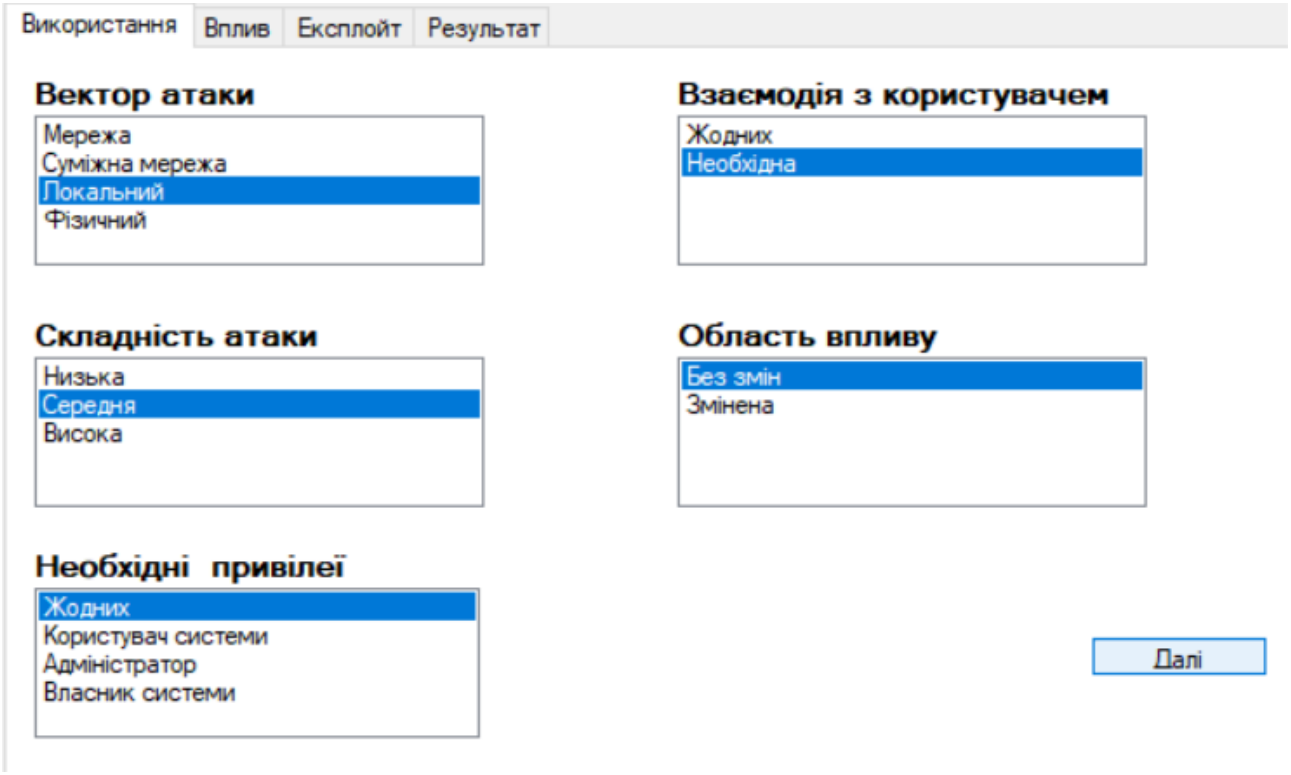

Рисунок 3.2 – Тестування компонента ListBox

При використанні компонента DataGridView (рис. 3.3), який надає можливість відображення та редагування табличних даних, можуть виникати наступні проблеми:

1. Некоректні дані: Якщо вхідні дані, які передаються до DataGridView, містять помилки або некоректні значення, це може призвести до неправильного відображення або непередбачуваної поведінки. Важливо перевірити та очистити дані перед їх використанням у DataGridView.

2. Проблеми з колонками та рядками: Неправильне налаштування колонок або рядків, таких як тип даних, ширина, порядок тощо, може призвести до некоректного відображення даних. Важливо належним чином налаштувати колонки та рядки DataGridView.

3. Проблеми з відображенням великої кількості даних: Якщо DataGridView використовується для відображення великої кількості даних, це може призвести до проблем з продуктивністю та відзначенням затримок при роботі з ним. Важливо оптимізувати відображення та роботу з великими обсягами даних.

4. Проблеми з сумісністю та версіями: У разі використання різних версій DataGridView або недостатньої сумісності з іншими компонентами або бібліотеками можуть виникати проблеми з функціональністю або некоректною роботою. Важливо

перевірити сумісність та налаштування всіх компонентів, що використовуються разом з DataGridView.

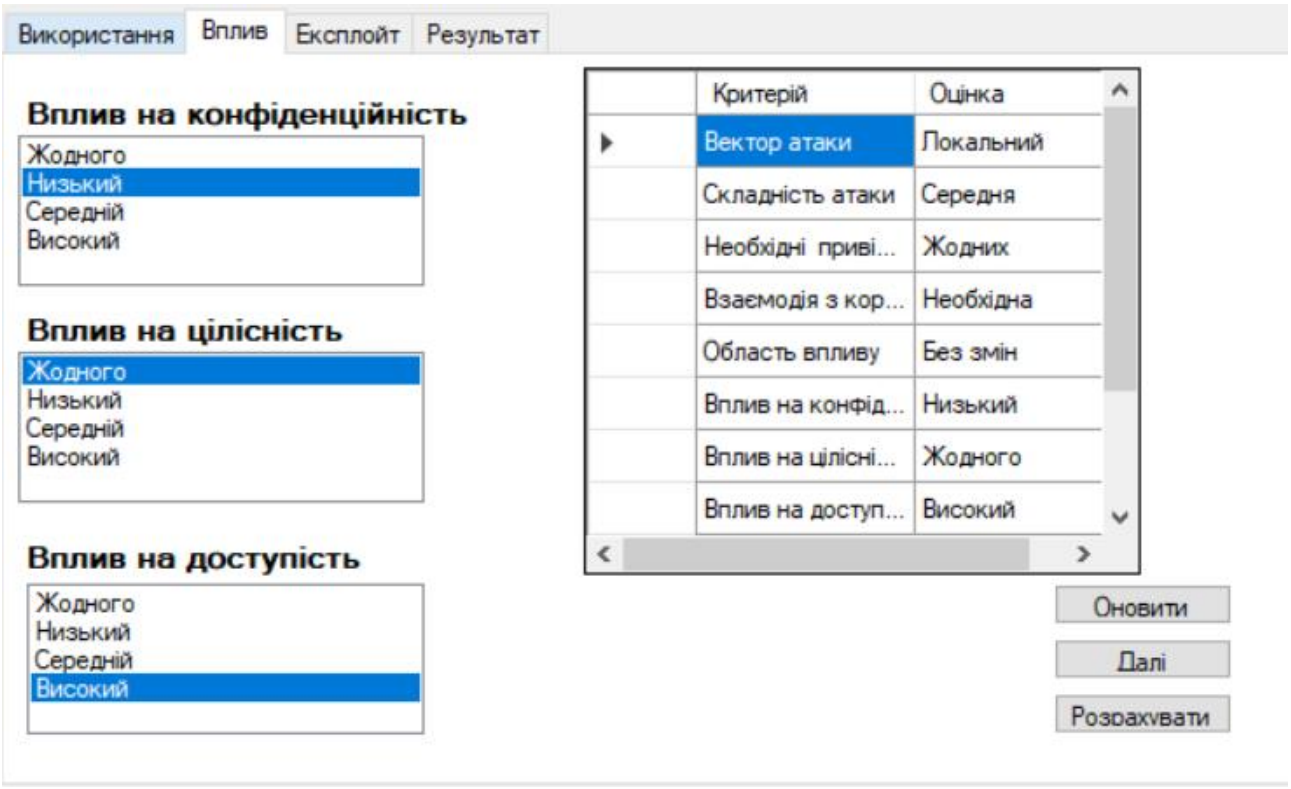

Рисунок 3.3 – Тестування компонента DataGridView

При використанні ChartBuilder (рис. 3.4), інструменту для створення графіків, можуть виникати наступні проблеми:

1. Некоректні дані: Якщо вхідні дані, передані в ChartBuilder, містять помилки або некоректні значення, це може призвести до неправильного відображення графіку або непередбачуваної поведінки програми. Важливо перевірити та очистити дані перед їх використанням у ChartBuilder.

2. Неправильне налаштування графіку: Якщо параметри графіку, такі як тип, масштаб, осі або колір, налаштовані неправильно, це може призвести до невірного відображення даних. Важливо перевірити правильність налаштувань графіку перед його відображенням.

3. Відсутність даних: Якщо вхідні дані для графіку відсутні або не вистачає даних для правильного побудови графіка, це може призвести до порожнього або неповного відображення графіку. Важливо перевірити, що всі необхідні дані доступні перед створенням графіка.

4. Неправильна інтерпретація графіку: Іноді графік може бути неправильно інтерпретований або незрозуміло відображений користувачем. Наприклад, осі можуть бути неправильно позначені або некоректно масштабовані. Важливо забезпечити зрозумілість та коректне відображення графіку для користувача.

5. Проблеми з сумісністю: Якщо ChartBuilder використовується в поєднанні з іншими компонентами або бібліотеками, можуть виникнути проблеми з їх сумісністю. Це може призвести до неправильної роботи графіків або навіть збоїв програми. Рекомендується перевірити сумісність та правильну налаштування всіх компонентів, що використовуються разом з ChartBuilder.

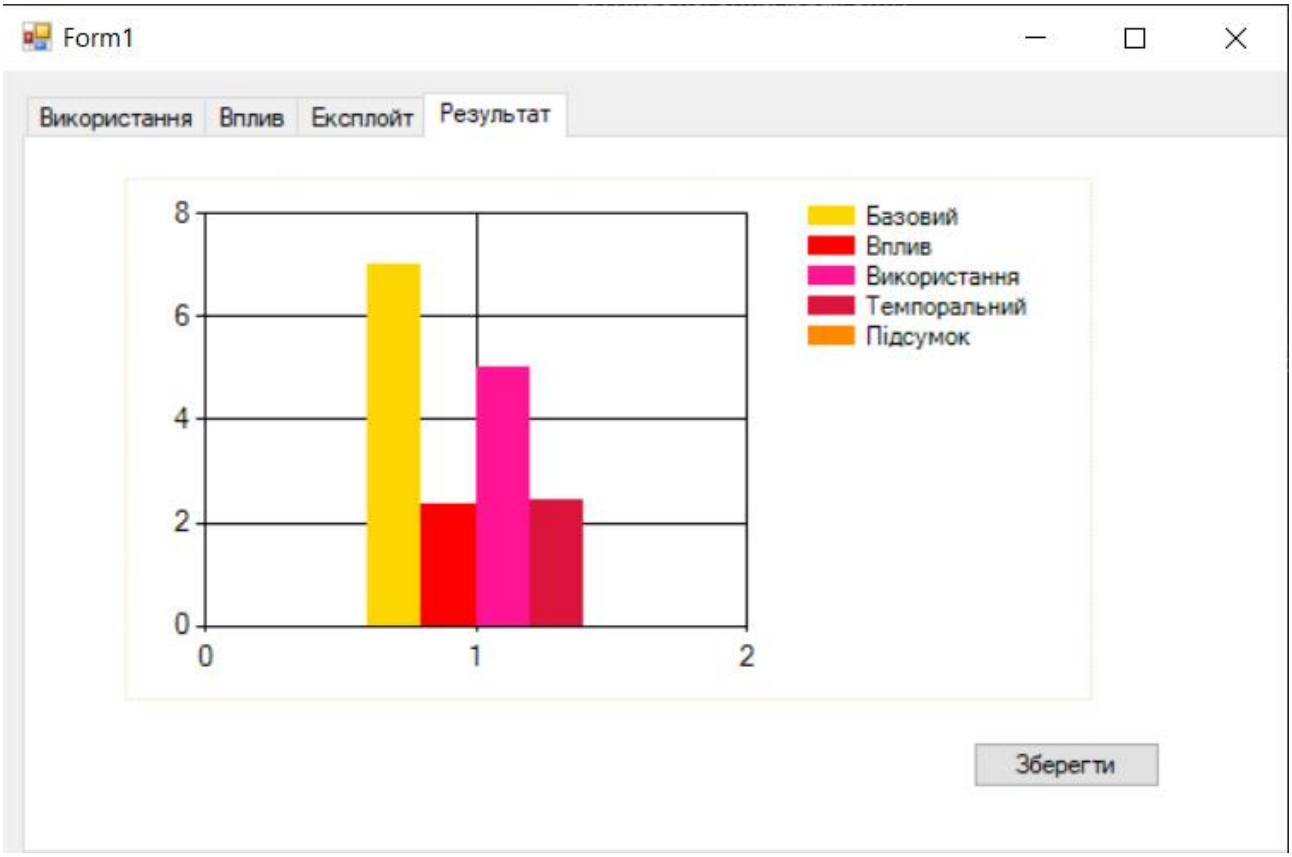

Рисунок 3.4 – Тестування компонента ChartBuilder

3.3. Рекомендації відносно зниження ризику виявлених загроз

Усі попередні кроки були спрямовані на виявлення найслабших областей в ІКС. Проте одним із важливих аспектів є зниження ризиків або попередження потенційних

атак. Нажаль перелік організацій у сфері кібербезпеки із міжнародним визнанням не дуже широкий. Він представлений такими варіантами, як:

1. Інформаційний центр з кібербезпеки (CERT): CERT-організації, такі як US-CERT, CERT-UK, CERT-In, відповідають за реагування на кібератаки та надання підтримки урядовим організаціям та критичним інфраструктурам. Вони забезпечують аналіз інцидентів, розробку рекомендацій з кібербезпеки та надання порад та підтримки у випадку кібернападів.

2. Інформаційно-аналітичні центри з кібербезпеки (SOC): SOC-організації забезпечують постійний моніторинг, виявлення та реагування на кібератаки. Вони аналізують дані з мережі та систем, виявляють незвичну активність та вживають заходів для захисту від кіберзагроз.

3. Кібербезпекові консалтингові компанії: Існують багато приватних компаній, які спеціалізуються на кібербезпеці та надають послуги консалтингу. Вони допомагають підприємствам розробляти та впроваджувати стратегії, політики та технології кібербезпеки, проводять аудити безпеки, виявляють вразливості та розробляють заходи для їх усунення.

4. Академічні дослідницькі лабораторії: Університети та дослідницькі інститути, такі як Mitre, CERT/CC в Карнегі-Меллон, Лабораторія кібербезпеки Університету Меріленда та інші, активно займаються дослідженнями в сфері кібербезпеки. Вони працюють над розробкою нових методів виявлення загроз, аналізу ризиків та захисту від кібератак.

Одною із систем, робота якої спеціалізується на цьому є Mitre. Mitre Corporation є некомерційною організацією, яка займається дослідженням і розвитком в різних галузях, включаючи науку, технології та системні рішення. Вона має значний досвід та експертизу в різних галузях, включаючи технології, науку, кібербезпеку, системну інженерію та інші. Використання їхніх знань дозволяє ефективно вирішувати складні технічні проблеми та впроваджувати інновації. Серед основних переваг можна виділити:

1. Незалежність: Mitre є некомерційною організацією, що дозволяє їй займатися об'єктивним дослідженням та розвитком без комерційних або політичних впливів.

2. Технічні знання: Mitre працює над широким спектром технологій та галузей, що дозволяє організації мати глибокі технічні знання в багатьох областях.

3. Співпраця з урядом: Mitre виконує багато проектів у співпраці з урядовими агентствами, зокрема з американським урядом. Це дає можливість Mitre вносити вагомий внесок у сферу національної безпеки, оборони та інших суспільно значущих справ.

4. Інновації: Mitre активно займається дослідженнями та розвитком нових технологій і рішень, спрямованих на вирішення складних проблем.

Звичайно попри всі переваги кожна система має свої недоліки, серед них:

1. Обмежене фінансування: Mitre фінансується переважно урядовими контрактами. Це може обмежувати їхню можливість залучати фінансування для окремих проектів або відмовлятися від проектів, які не відповідають урядовим потребам.

2. Обмеженість впровадження: Оскільки Mitre не є комерційною організацією, їхні інноваційні рішення можуть не отримувати широкого впровадження на ринку через відсутність комерційних структур і каналів продажу.

3. Конфіденційність інформації: Оскільки Mitre співпрацює з урядом, вони можуть бути обмежені в розкритті деякої інформації або результатів своїх досліджень через конфіденційність та безпеку національних проектів.

Mitre займається розробкою та впровадженням інноваційних методів та рішень для захисту критичних інформаційних систем від кібератак. Вони досліджують нові загрози, розробляють стратегії кібербезпеки та пропонують рекомендації для підвищення рівня захисту. Вони працюють над розробкою нових методів штучного інтелекту, машинного навчання, квантових обчислень та інших передових технологій. Саме тому їх база є одним із кращих варіантів для надання рекомендацій (рис. 3.5). Така інтеграція дозволить значно полегшити подальшу розробку стратегії захисту досліджуваної ІКС.

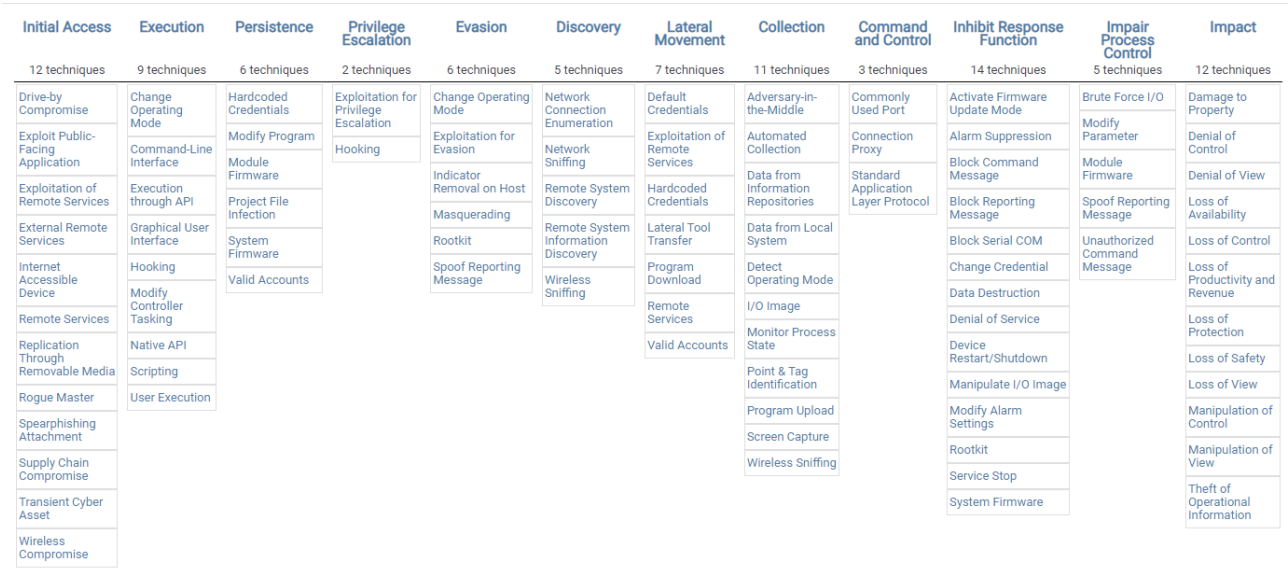

Рисунок 3.5 – Бібліотека Mitre

Висновки за розділом 3

У даній програмній реалізації було розроблено додаток, що використовує компоненти Windows Forms. Додаток дозволяє користувачеві вводити дані за допомогою ListBox і відображати їх у вигляді табличної структури за допомогою DataGridView, виконувати основні розрахунки та відображати їх у вигляді двограм за допомогою компонента ChartBuilder.

Основні функції додатку включають:

1. Введення даних: Користувач може вводити дані, вибираючи їх з ListBox. Дані відображаються у відповідних колонках DataGridView.

2. Відображення даних: Введені дані відображаються у вигляді табличної структури в DataGridView, що забезпечує зручний та організований перегляд даних.

3. Оновлення даних: У разі необхідності корегування даних можливо обрати інший варіант у ListBox та скористатися кнопкою «Оновити» для перегляду актуальних даних.

4. Збереження: Всі дані та результати можна зберегти з розширеннями файлів, такими як CSV або Excel.

Переваги цієї програмної реалізації включають:

- Використання готових компонентів Windows Forms спрощує розробку і забезпечує швидку інтеграцію з іншими компонентами.
- Використання ListBox дозволяє легко вводити дані шляхом вибору зі списку, що полегшує користувачеві введення даних.
- Використання DataGridView дозволяє зручно відображати та редагувати табличні дані.
- Перевірка на валідність введених даних, наприклад, перевірка на наявність обов'язкових полів.
- Реалізація функцій збереження та завантаження даних з розширеннями файлів, такими як CSV або Excel, для забезпечення зручного обміну даними.

У цілому, ця програмна реалізація демонструє можливості використання компонентів для введення, обробки, відображення та збереження даних у додатку Windows Forms. Вона забезпечує зручний та організований спосіб роботи з даними, але може бути доповнена додатковими функціями та вдосконаленнями для покращення користувацького досвіду.

ВИСНОВКИ

У даній кваліфікаційній роботі була проведена оцінка захищеності інформаційно-комунікаційної системи на основі інтелектуальних технологій. Досліджено різні аспекти безпеки, включаючи виявлення вразливостей, ризик-аналіз та заходи захисту. Робота спрямована на покращення безпеки інформаційних систем та забезпечення захисту від потенційних загроз.

У першому розділі було розглянуто теоретичні аспекти оцінки захищеності системи. Було вивчено методи інтелектуального аналізу даних, а також інструменти та технології, які можна використовувати для оцінки безпеки системи. Зазначено, що використання інтелектуальних технологій може значно полегшити процес оцінки та підвищити ефективність заходів захисту.

У другому розділі була проведена аналітична робота, включаючи збір і аналіз даних про методології. Використовуючи різні методи та алгоритми, були ідентифіковані слабкі місця різних підходів, які потребують підвищеної уваги та покращення. Проведений аналіз дозволяє розширити спектр критеріїв та рівня їх впливу для отримання більш точних результатів та розробити методику оцінки захищеності системи на основі отриманих даних. Для цього використовувалися показники, що враховують різні аспекти безпеки, такі як ідентифіковані загрози, рівень вразливостей та ефективність заходів захисту. Запропонована методика дозволяє чисельно оцінити рівень безпеки системи та виявити потенційні проблеми.

У третьому розділі розрахункові формули та алгоритми були описані та дороблені для визначення рівня захищеності системи із урахуванням нововведень. А також була розроблена програмна реалізація для оцінки захищеності системи у вигляді додатку з використанням компонентів ListBox, DataGridView та ChartBuilder, які дозволяють зручно вводити дані та відображати результати оцінки. Програма також надає можливість збереження результатів оцінки та їх експорту у формат Excel для подальшого аналізу. При розробці програми було враховано принципи інтерфейсу користувача та забезпечено його зручність та інтуїтивність.

В ході дослідження було виявлено, що використання інтелектуальних технологій дозволяє ефективно виявляти вразливості та ризики, а також розробляти ефективні заходи захисту.

Отже, результати дослідження та розробки, проведені в даній дипломній роботі, дають цінні висновки та рекомендації щодо покращення захищеності інформаційно-комунікаційних систем на основі інтелектуальних технологій. Дані результати можуть бути використані для подальших досліджень та впровадження в практичну діяльність організацій для забезпечення безпеки їх інформаційних систем.

Результати досліджень були апробовані на VI міжнародній науково-практичній конференції «Проблеми кібербезпеки інформаційно-телекомунікаційних систем» (PCSITS).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хорошко В. О., Хохлачова Ю. Є. Оцінка захищеності інформаційних систем – 2012.
2. Будько М.М. Методи оцінки загроз для інформації автоматизованих систем. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні – 2005. – Вип. 10.
3. Андреєв В.І., Хорошко В.О., Чередниченко В.С., Шелест М.Є. Основи інформаційної безпеки – 2009.
4. Еколого-економічний ризик-менеджмент: методи оцінювання ризиків: Навч. посібник Національного технічного університету Київського політехнічного інституту імені Ігоря Сікорського // Караєва Н. В. – 2019.
5. ДСТУ 3396.2-97 Захист інформації. Технічний захист інформації. Терміни та визначення / Державний стандарт України.
6. ДСТУ ISO/IEC 27005:2015 Інформаційні технології. Методи захисту. Управління ризиками інформаційної безпеки / Державний стандарт України.
7. ДСТУ IEC/ISO 31010:2013 Керування ризиком. Методи загального оцінювання ризику (IEC/ISO 31010:2009, IDT)/ Державний стандарт України.
8. Rebecca M. Blank, Acting Secretary, Patrick D. Gallagher NIST Special Publication 800-30 Revision 1, Guide for Conducting Risk Assessments – 2012.
9. Інтелектуальні інформаційні системи: Навч. посібник // Шаров С.В., Лубко Д.В., Осадчий В.В. – 2015.
10. Застосування інтелектуальних технологій для підвищення якості роботи телекомунікаційних мереж при невизначеності: Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка // Селюков О. В., Хмельницький Ю. В., Обертюк І. В., Солодєєва Л. В. –2017. – Вип. 56.
11. ДСТУ 2481-94 Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення / Державний стандарт України.
12. Biffl Stefan, Sabou Marta Semantic Web Technologies for Intelligent Engineering Applications. – Springer, 2016.

13. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу [Електронний ресурс]. – Режим доступу: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?showHidden=1&art_id=101870&cat_id=89734&ctime=1344501089407

14. Аудит безпеки інформаційної безпеки [Електронний ресурс]. – Режим доступу: https://tzi.com.ua/audbezib.html

15. Піскун С.Ж., Хорошко В.О., Хохлачова Ю.Є. Оцінка безпеки інформаційної сфери. Сучасна спеціальна техніка. – 2013.

16. Методика оцінки рівня безпеки інформації: Вісник НУ «Львівська політехніка» // Габович А.Г., Горобець А.Ю., Горобець А.Ю., Хорошко В.О. – 2006.

17. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах / Постанова КМ України від 29.03.2006 № 373.

18. НД ТЗІ 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу, затверджений наказом ДСТСЗІ СБ України від 28.04.1999 № 22.

19. Андреєв В.І., Хорошко В.О., Чередниченко В.С., Шелест М.Є. Основи інформаційної безпеки – 2009.

20. НД ТЗІ 2.5-004-99 Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу, затверджений наказом ДСТСЗІ СБ України від 28.04.1999 № 22.

21. Сірченко Г.А., Хорошко В.О., Хохлачова Ю.Є. Алгоритм визначення показників для оцінки надійності систем спеціального призначення. – 2013.

22. Вертузаєв М.С. , Юрченко О. М. Захист інформації в комп'ютерних системах від несанкціонованого доступу – 2001.

23. ДСТУ ISO 31000:2018 Менеджмент ризиків. Принципи та настанови (ISO 31000:2018, IDT)/ Державний стандарт України.

24. Ніколаюк С.І., Никифорчук Д.Й., Томма Р.П., Барко В.І. Протидія злочинам у сфері інтелектуальної власності. – 2006.

25. Виявлення та розслідування злочинів, що вичиняються у сфері інформаційних технологій: Наук.-практ.посіб.// За заг.ред.проф. Я.Ю.Кондратьєва. – 2004.

ДОДАТОК А

```
System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp2
{
   public partial class Form1 : Form
   {
      public string[] Labels = new string[11];
      public string[] columns = { "Критерій", "Оцінка" };
      public string[] items = new string[11];
      public double ISCbase = 0;
      public double Base_Impact = 0;
      public double Base = 0;
      public double Temporal;
      public Form1()
      {
         InitializeComponent();
         Labels.SetValue("Вектор атаки", 0);
         Labels.SetValue("Складність атаки", 1);
         Labels.SetValue("Необхідні  привілеї", 2);
         Labels.SetValue("Взаємодія з користувачем", 3);
         Labels.SetValue("Область впливу", 4);
         chart1.Series.Clear();
         chart1.Series.Add("Базовий");
         chart1.Series.Add("Вплив");
         chart1.Series.Add("Використання");
         chart1.Series.Add("Темпоральний");
         chart1.Series.Add("Підсумок");
         //Labels.Append(label1.Text, label2.Text, label3.Text, label4.Text, label5.Text, label6.Text,
label7.Text, label8.Text, label9.Text, label10.Text, label11.Text ).ToArray();
         string tip1 = "Цей показник відображає контекст, у якому можливе використання
вразливостей.";
         toolTip1.SetToolTip(label1, tip1);
         string tip2 = "Цей показник описує умови поза контролем зловмисника, які мають існувати
для використання вразливості.";
         toolTip2.SetToolTip(label2, tip2);
         string tip3 = "Цей показник характеризує рівень необхідних привілеїв для успішного
проведення атаки.";
         toolTip2.SetToolTip(label3, tip3);
         string tip4 = "Цей показник характеризує вимоги до користувача для успішної
компрометації компонента.";
         toolTip4.SetToolTip(label4, tip4);
```

```
        string tip5 = "Цей показник характеризує здатність уразливості в одному із компонентів програмного забезпечення впливати на область поза його можливостями або привілеями.";
        toolTip5.SetToolTip(label5, tip5);
        string tip6 = "Цей показник вимірює вплив на конфіденційність інформаційних ресурсів, якими керує програмний компонент через використану вразливість.";
        toolTip6.SetToolTip(label6, tip6);
        string tip7 = "Цей показник вимірює вплив на цілісність інформаційних ресурсів. Цілісність означає достовірність та правдивість інформації.";
        toolTip7.SetToolTip(label7, tip7);
        string tip8 = "Цей показник вимірює вплив успішно використаної вразливості на доступність відповідного компонента.";
        toolTip8.SetToolTip(label8, tip8);
        string tip9 = "Цей показник вимірює ймовірність атаки на вразливість і базується на поточному стані використання методів, доступності використання коду або активності використання.";
        toolTip9.SetToolTip(label9, tip9);
        string tip10 = "Цей показник визначає пріоритетність, коригує часовий бал у бік зменшення, відображаючи зниження терміновості остаточного рішення.";
        toolTip10.SetToolTip(label10, tip10);
        string tip11 = "Цей показник вимірює ступінь впевненості в існуванні вразливості та достовірності відомості технічних деталей.";
        toolTip11.SetToolTip(label11, tip11);

        string[] list1 = { "Мережа", "Суміжна мережа", "Локальний", "Фізичний" };
        for (int i=0; i<list1.Length; i++)
        {
            listBox1.Items.Add(list1[i]);
        }
        string[] list2 = { "Низька", "Середня", "Висока" };
        for (int i = 0; i < list2.Length; i++)
        {
            listBox2.Items.Add(list2[i]);
        }
        string[] list3 = { "Жодних", "Користувач системи", "Адміністратор", "Власник системи" };
        for (int i = 0; i < list3.Length; i++)
        {
            listBox3.Items.Add(list3[i]);
        }
        string[] list4 = { "Жодних", "Необхідна" };
        for (int i = 0; i < list4.Length; i++)
        {
            listBox4.Items.Add(list4[i]);
        }
        string[] list5 = { "Без змін", "Змінена" };
        for (int i = 0; i < list5.Length; i++)
        {
            listBox5.Items.Add(list5[i]);
        }
        string[] list6 = { "Жодного", "Низький", "Середній", "Високий" };
        for (int i = 0; i < list6.Length; i++)
        {
            listBox6.Items.Add(list6[i]);
```

```
            listBox7.Items.Add(list6[i]);
            listBox8.Items.Add(list6[i]);
        }
        string[] list9 = { "Не визначено", "Не доведено існування", "Підтверджена концепція",
"Функціональний експлойт", "Висока" };
        for (int i = 0; i < list9.Length; i++)
        {
            listBox9.Items.Add(list9[i]);
        }
        string[] list10 = { "Не визначено", "Офіційне виправлення", "Тимчасове виправлення",
"Обхідний шлях", "Недоступне" };
        for (int i = 0; i < list10.Length; i++)
        {
            listBox10.Items.Add(list10[i]);
        }
        string[] list11 = { "Не визначено", "Невідомо", "Достовірно", "Підтверджено" };
        for (int i = 0; i < list11.Length; i++)
        {
            listBox11.Items.Add(list11[i]);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        dataGridView1.Columns.Clear();
        dataGridView1.Rows.Clear();
        items.SetValue(listBox1.Text, 0);
        items.SetValue(listBox2.Text, 1);
        items.SetValue(listBox3.Text, 2);
        items.SetValue(listBox4.Text, 3);
        items.SetValue(listBox5.Text, 4);
        for (int i = 0; i < columns.Length; i++)
        {
            dataGridView1.Columns.Add("", columns[i]);
        }
        for (int i = 0; i < Labels.Length; i++)
        {
            dataGridView1.Rows.Add(Labels[i], items[i]);
        }
        tabControl1.SelectedIndex = 1;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        dataGridView1.Columns.Clear();
        dataGridView1.Rows.Clear();
        items.SetValue(listBox6.Text, 5);
        items.SetValue(listBox7.Text, 6);
        items.SetValue(listBox8.Text, 7);
        Labels.SetValue("Вплив на конфіденційність",5);
        Labels.SetValue("Вплив на цілісність", 6);
        Labels.SetValue("Вплив на доступність", 7);
```

```
    for (int i = 0; i < columns.Length; i++)
    {
        dataGridView1.Columns.Add("", columns[i]);
    }
    for (int i = 0; i < Labels.Length; i++)
    {
        dataGridView1.Rows.Add(Labels[i], items[i]);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    dataGridView2.Columns.Clear();
    dataGridView2.Rows.Clear();
    tabControl1.SelectedIndex = 2;
    for (int i = 0; i < columns.Length; i++)
    {
        dataGridView2.Columns.Add("", columns[i]);
    }
    for (int i = 0; i < Labels.Length; i++)
    {
        dataGridView2.Rows.Add(Labels[i], items[i]);
    }
}

private void button4_Click(object sender, EventArgs e)
{
    dataGridView2.Columns.Clear();
    dataGridView2.Rows.Clear();
    items.SetValue(listBox9.Text, 8);
    items.SetValue(listBox10.Text, 9);
    items.SetValue(listBox11.Text, 10);
    Labels.SetValue("Зрілість коду експлойту", 8);
    Labels.SetValue("Рівень виправлення", 9);
    Labels.SetValue("Достовірність звіту", 10);
    for (int i = 0; i < columns.Length; i++)
    {
        dataGridView2.Columns.Add("", columns[i]);
    }
    for (int i = 0; i < Labels.Length; i++)
    {
        dataGridView2.Rows.Add(Labels[i], items[i]);
    }
}
public void Summary()
{
    chart1.Series["Базовий"].ResetIsValueShownAsLabel();
    chart1.Series["Вплив"].ResetIsValueShownAsLabel();
    chart1.Series["Використання"].ResetIsValueShownAsLabel();
    chart1.Series["Темпоральний"].ResetIsValueShownAsLabel();
    chart1.Series["Підсумок"].ResetIsValueShownAsLabel();
```

```
        double vector=0, complexity=0, privileges=0, user=0, imp_area=0, imp_conf=0, imp_intag=0,
imp_avail=0, maturity=0, level=0, report = 0;
        if (listBox1.SelectedIndex == 0)
        {
           vector = 1;
        }
        else if (listBox1.SelectedIndex == 1)
        {
           vector = 0.646;
        }
        else if(listBox1.SelectedIndex == 2)
        {
           vector = 0.521;
        }
        else if(listBox1.SelectedIndex == 3)
        {
           vector = 0.395;
        }
        else if (listBox1.SelectedItems.Count == 0)
        {
           errorProvider1.SetError(button6,"Check data");
        }
        if (listBox2.SelectedIndex == 0)
        {
           complexity = 0.35;
        }
        else if (listBox2.SelectedIndex == 1)
        {
           complexity = 0.48;
        }
        else if (listBox2.SelectedIndex == 2)
        {
           complexity = 0.61;
        }
        else if (listBox2.SelectedItems.Count == 0)
        {
           errorProvider1.SetError(button6, "Check data");

        }
        if (listBox3.SelectedIndex == 0)
        {
           privileges = 0.96;
        }
        else if (listBox3.SelectedIndex == 1)
        {
           privileges = 0.85;
        }
        else if (listBox3.SelectedIndex == 2)
        {
           privileges = 0.62;
        }
        else if (listBox3.SelectedIndex == 3)
```

```
{
   privileges = 0.27;
}
else if (listBox3.SelectedItems.Count == 0)
{
   errorProvider1.SetError(button6, "Check data");
}
if (listBox4.SelectedIndex == 0)
{
   user = 0.85;
}
else if (listBox4.SelectedIndex == 1)
{
   user = 0.62;
}
else if (listBox4.SelectedItems.Count == 0)
{
   errorProvider1.SetError(button6, "Check data");
}
if (listBox5.SelectedIndex == 0)
{
   imp_area = 0;
   //Base_Impact = 6.42 * ISCbase;
}
else if (listBox5.SelectedIndex == 1)
{
   imp_area = 10;
   //Base_Impact =
}
else if (listBox5.SelectedItems.Count == 0)
{
   errorProvider1.SetError(button6, "Check data");

}
if (listBox6.SelectedIndex == 0)
{
   imp_conf = 0;
}
else if (listBox6.SelectedIndex == 1)
{
   imp_conf = 0.4;
}
else if (listBox6.SelectedIndex == 2)
{
   imp_conf = 0.7;
}
else if (listBox6.SelectedIndex == 3)
{
   imp_conf = 1;
}
else if (listBox6.SelectedItems.Count == 0)
{
```

```
    errorProvider1.SetError(button6, "Check data");
}
if (listBox7.SelectedIndex == 0)
{
    imp_intag = 0;
}
else if (listBox7.SelectedIndex == 1)
{
    imp_intag = 0.4;
}
else if (listBox7.SelectedIndex == 2)
{
    imp_intag = 0.7;
}
else if (listBox7.SelectedIndex == 3)
{
    imp_intag = 1;
}
else if (listBox7.SelectedItems.Count == 0)
{
    errorProvider1.SetError(button6, "Check data");
    errorProvider2.SetError(label7, "Check data");
}
if (listBox8.SelectedIndex == 0)
{
    imp_avail = 0;
}
else if (listBox8.SelectedIndex == 1)
{
    imp_avail = 0.4;
}
else if (listBox8.SelectedIndex == 2)
{
    imp_avail = 0.7;
}
else if (listBox8.SelectedIndex == 3)
{
    imp_avail = 1;
}
else if (listBox8.SelectedItems.Count == 0)
{
    errorProvider1.SetError(button6, "Check data");
}
if (listBox9.SelectedIndex == 0)
{
    maturity = 0;
}
else if (listBox9.SelectedIndex == 1)
{
    maturity = 0.3;
}
else if (listBox9.SelectedIndex == 2)
```

```
{
   maturity = 0.5;
}
else if (listBox9.SelectedIndex == 3)
{
   maturity = 0.7;
}
else if (listBox9.SelectedIndex == 4)
{
   maturity = 1;
}
if (listBox10.SelectedIndex == 0)
{
   level = 0;
}
else if (listBox10.SelectedIndex == 1)
{
   level = 0.3;
}
else if (listBox10.SelectedIndex == 2)
{
   level = 0.5;
}
else if (listBox10.SelectedIndex == 3)
{
   level = 0.7;
}
else if (listBox10.SelectedIndex == 4)
{
   level = 1;
}
if (listBox11.SelectedIndex == 0)
{
   report = 0;
}
else if (listBox11.SelectedIndex == 1)
{
   report = 0.3;
}
else if (listBox11.SelectedIndex == 2)
{
   report = 0.7;
}
else if (listBox11.SelectedIndex == 3)
{
   report = 1;
}
double Base_Exploitability = (8.22* vector * complexity * user * privileges);
ISCbase = 1 - ((1 - imp_conf) * (1 - imp_intag) * (1 - imp_avail));
//double Base_Impact = ((imp_area * (imp_conf + imp_intag + imp_avail)) * 0.029) - 3.25;
if(imp_area==0)
{
```

```
            Base_Impact = 6.42 * ISCbase;
            Base = Math.Round(Math.Min((Base_Impact + Base_Exploitability), 10));
            Temporal = Base * maturity * level * report;
        }
        else
        {
            Base_Impact = 7.52 * (ISCbase - 0.029) - 3.25 * (ISCbase - 0.02);
            Base = Math.Round(Math.Min(1.08*(Base_Impact + Base_Exploitability), 10));
            Temporal = Base * maturity * level * report;
        }
        chart1.Series["Базовий"].Points.AddY(Base);
        chart1.Series["Вплив"].Points.AddY(Base_Exploitability);
        chart1.Series["Використання"].Points.AddY(Base_Impact);
        chart1.Series["Темпоральний"].Points.AddY(Temporal);
    }
    private void button6_Click(object sender, EventArgs e)
    {
        Summary();
        tabControl1.SelectedIndex = 3;
    }
private void button7_Click(object sender, EventArgs e)
    {
        XSSFWorkbook CVSS = new XSSFWorkbook();
        ISheet worksheet = CVSS.CreateSheet("Sheet1");
        for (int i = 0; i < Labels.Length; i++)
        {
            IRow row = worksheet.CreateRow(i);
            row.CreateCell(0).SetCellValue(Labels[i]);
            row.CreateCell(1).SetCellValue(items[i]);
        }
        IRow res = worksheet.CreateRow(Labels.Length);
        res.CreateCell(0).SetCellValue("Підсумок");
        res.CreateCell(1).SetCellValue(Base);


        using (FileStream stream = new FileStream("output.xlsx", FileMode.Create, FileAccess.Write))
        {
            CVSS.Write(stream);
        }
    }
```