

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи захисту мережевої ІТ-
інфраструктури на об’єктах критичного застосування”**

Виконав здобувач вищої освіти
IV курсу, групи КІ-23-МБ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Толмачов Я.П.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань ._12 _____“Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Толмачову Ярославу Петровичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи захисту мережевої
ІТ-інфраструктури на об’єктах критичного
застосування*

2. Керівник роботи *Якименко Наталія Миколаївна, кандидат фізико-математичних
наук, доцент*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 126-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка
програмного забезпечення системи захисту мережевої ІТ-інфраструктури на
об’єктах критичного застосування*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити)

1. Призначення та область використання.

2.Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Якименко Н.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Толмачов Я.П.
(прізвище та ініціали)

АНОТАЦІЯ

Толмачов Я.П. Програмне забезпечення системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

Метою розробки є програмне забезпечення системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

Результат роботи – програмна реалізація системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, мережева ІТ-інфраструктура, об'єкти критичного застосування

ABSTRACT

Tolmachov Ya.P. Software for the protection system of network IT infrastructure at critical application facilities. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the protection system of network IT infrastructure at critical application facilities.

The purpose of the development is the software for the protection system of network IT infrastructure at critical application facilities.

The result of the work is the software implementation of the protection system of network IT infrastructure at critical application facilities.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on PCs with Windows 10/11.

The program is developed in the Python environment.

Keywords: computer engineering, network IT infrastructure, critical application objects

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	25
2.3 Розгорнута постановка завдання	30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	31
3.1 Опис функціонування системи	31
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	48
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	50
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	50
4.2 Захист розробленого програмного забезпечення.....	74
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	77
6 ОСНОВНІ ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86

					ВКРБ-123.26.0036.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи захисту мережевої IT-інфраструктури на об'єктах критичного застосування	Літ.	Аркуш	Аркушів
Розроб.	Толмачов Я.П.					Б	1	94
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23-МБ		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІТ	—	інформаційні технології
КІІ	—	критична інформаційна інфраструктура
НС	—	надзвичайна ситуація
ОС	—	операційна система
ПЗ	—	програмне забезпечення
CERT-UA	—	урядова команда реагування на комп'ютерні надзвичайні події України
CSIRT	—	група реагування на інциденти, пов'язані з комп'ютерною безпекою
СІІ	—	Critical Information Infrastructure – критична інформаційна інфраструктура
СІІР	—	Critical Information Infrastructure Protection
NASK	—	Research and Academic Computer Network – дослідницька й академічна комп'ютерна мережа
NCSC	—	National Cyber Security Centre – національний центр кібербезпеки
SAMFI	—	Cooperation Group for Information Security – шведська група співробітництва в області інформаційної безпеки
QoS	—	механізми контролю й керування якістю

ВСТУП

Актуальність теми. Використовуване на багатьох підприємствах ПЗ застаріло й буквально «поїдене» уразливостями. Для поліпшення ситуації ІТ-інфраструктуру необхідно переводити на вітчизняні рішення.

Технологічна пара «процесор – операційна система» – фундамент ІТ-інфраструктури будь-якої організації, незалежно від її галузевої приналежності, масштабу й складу апаратно-програмних засобів. На багатьох підприємствах використовуються версії програмного забезпечення, які давно не підтримуються виробником і буквально «поїдені» уразливостями.

Проблема збільшується можливою наявністю в програмному продукті так званих закладок, за допомогою яких дані, що представляють певну цінність, могли протягом багатьох років передаватися стороннім особам. Крім того, закладки несуть погрозу несанкціонованого віддаленого керування: тільки представте, що в якусь «годину Х» імпортна ОС, що контролює роботу встаткування АЕС, виявляється під керуванням недружньої держави або організації. Це справжня бомба вповільненої дії.

Зовсім очевидно, що ІТ-інфраструктуру критично важливих об'єктів треба перебудовувати з урахуванням вимог інформаційної безпеки, поетапно переводячи її на вітчизняні рішення. Першим кроком може стати заміна ОС Windows вітчизняною операційною системою, здатною підтримувати роботу не тільки Linux-версій прикладного ПЗ, але й успадкованих Windows-версій такого ПЗ. Однак надто важливо, щоб ця система була дійсно українською.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

- Огляд існуючих систем захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.
- Дослідження системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.
- Програмна реалізація системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У чому ключова відмінність ОС, розвиток якого управляється з юрисдикції України, від аналогічного рішення, виконаного на базі закордонного дистрибутива? Будь-яка операційна система розвивається на базі репозиторія – свого роду «фабрики» по виробництву ПЗ, у якій об'єднані інфраструктури розробки, зберігання, модернізації й реалізації життєвого циклу програмного забезпечення. ОС, створювана на базі закордонного репозиторія, завжди є залежним від життєвого циклу цього репозиторія. Зокрема, це стосується виходу нових і підтримки застарілих версій. Зміни, які вітчизняні розроблювачі вносять у таку ОС в інтересах українських користувачів, у більшості випадків не включаються в репозиторій. Тому вони змушені організовувати власну вітку розробки й самостійно забезпечувати сумісність своїх нововведень із актуальною платформою закордонного репозиторія. Це сильно гальмує вихід нових версій української ОС, оскільки всі наробітки доводиться заново вбудовувати у свіжу редакцію закордонного програмного продукту. Так відбувається, наприклад, з компонентами, що забезпечують сумісність операційної системи з українськими апаратними. Раніше написані прикладні програми далеко не завжди коректно функціонують у новій ОС і мають потребу в додатковій переробці.

Операційна система, життєвий цикл якої повністю реалізується на вітчизняній технологічній базі, урятована від цих погроз. Надто важливо, що перехід на таку систему можна здійснити поступово, оскільки в неї убудовані системні механізми, що дозволяють комп'ютерам під керуванням Windows і Linux працювати в єдиному гетерогенному середовищі. Крім того, така ОС здатна управляти цифровою інфраструктурою практично необмеженого масштабу й складності.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Це означає, що організація може здійснити перехід на вітчизняні технології еволюційно, зберігаючи інвестиції, вкладені в інфраструктуру, нарощуючи її безпеку й опираючись на гарантовану якісну технічну підтримку, надавану на всій території країни.

1.2 Область застосування

Ризики для критично важливої інфраструктури, такої як енергетика, водопостачання та цифрові системи, на мою думку, часто недооцінюються. Люди ігнорують ці ризики, доки серйозний інцидент не потрапить у заголовки газет.

Останній звіт про глобальні ризики демонструє певну обізнаність щодо цих зростаючих викликів, але багато вразливостей все ще залишаються непоміченими. В опитуванні експертів «перебої в критичній інфраструктурі» посіли лише 22-ге місце протягом наступних двох років і 23-тє протягом наступних десяти років. Це разюче низький показник для чогось настільки фундаментального для сучасного життя. Я боюся, що це небезпечний недогляд: коли інфраструктура виходить з ладу, системи, які від неї залежать, також можуть вийти з ладу. Інфраструктура також залишається головною ціллю в конфлікті, посідаючи п'яте місце протягом наступних двох років. Підводні комунікаційні кабелі можуть бути перерізані; супутники можуть бути глушені; дрони можуть загрожувати критично важливим об'єктам. Доступ до життєво важливих ресурсів – їжі, води, енергії – може бути використаний як зброя в геополітичних суперечках.

Цифровізація перетворює це на гібридну загрозу, а кібербезпека також увійде до 10 головних ризиків протягом наступного десятиліття. Підключена інфраструктура, така як мережі, порти та трубопроводи, пропонує ширшу поверхню для атак як для злочинних груп, так і для пов'язаних з державою суб'єктів. Наприклад, успішна кібератака на енергетичну систему або мережу управління повітряним рухом може мати каскадні наслідки, що виходять далеко

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

за межі початкової точки збою, посилюючи інші фізичні або геополітичні потрясіння. Частина проблеми полягає в нашому сприйнятті ризику. Наша зосередженість на безпосередній кризі може призвести до того, що ми відкладемо превентивні заходи (такі як технічне обслуговування) ще на рік. Погляди занадто часто розрізняються. Наприклад, кліматична та фінансова стійкість іноді розглядаються як окремі питання та вирішуються за допомогою різних стратегій. Ризики, які сприймаються як більш драматичні – конфлікти, пандемії, екстремальні погодні умови – часто відтісняють менш очевидні, але не менш критичні загрози з поля зору. Такий фрагментарний підхід не враховує фундаментальної реальності: спектр глобальних ризиків взаємопов'язаний. Проблема, з якою стикається критична інфраструктура, підкреслює це. Геополітична напруженість, технологічні збої та соціальна нестабільність – все це сходиться навколо тих самих базових систем. Електромережа, що перебуває під навантаженням через спеку, також може бути вразливою до кібератак. Порт, що постраждав від повені, може посилити шоки в ланцюжку поставок та соціальні заворушення. Слабкі державні фінанси обмежують можливості пом'якшення або реагування на будь-яку з цих загроз. В епоху конкуренції наш інстинкт може спонукати нас зосередитися лише на останній кризі. Я вважаю, що нагальним викликом – і водночас можливістю – для урядів та осіб, які приймають рішення, є навпаки: розширити об'єктив, побачити, як ризики підсилюють один одного, та розробити рішення подвійного або багатоцільового використання, які одночасно зміцнюють кілька вимірів стійкості. Лідери повинні діяти заздалегідь, щоб зміцнити основи, на яких ґрунтується стабільність наших економік і суспільств. Критична інфраструктура є критичним ризиком. Настав час ставитися до неї відповідно.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Критично важливий об'єкт інфраструктури – це такий об'єкт, порушення або припинення працездатності якого приводить до необоротних негативних змін в економіці адміністративно-територіальної одиниці або всієї країни, а також до істотного погіршення безпеки населення, що проживає на цій території.

У критичну інфраструктуру входить 12 категорій:

- охорона здоров'я;
- наука;
- транспорт;
- зв'язок;
- атомна енергетика;
- паливно-енергетичний комплекс;
- банківська сфера;
- оборонна сфера;
- ракетно-космічна сфера;
- гірничодобувна промисловість;
- металургійна промисловість;
- хімічна промисловість.

Надійний захист настільки важливих об'єктів на перший погляд може здатися неймовірно складним завданням. Звичайно, він складний, однак стає цілком посильним, якщо розбити його на 5 послідовних кроків:

- Визначити потенційні погрози.
- Побудувати концепцію безпеки.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- Зробити вибір на користь інтегрованої системи.
- Віддати перевагу системі, що використовує нейроаналітику.
- Вибрати правильну інтегруючу платформу.

Крок 1. Визначити потенційні погрози

Основні потенційні погрози для об'єктів критичної інфраструктури можна розділити на 3 групи:

- **Зовнішні людські фактори.** Тероризм, бандитизм і інші дії кримінального характеру (підриви, підпали, використання небезпечних речовин).
- **Внутрішні людські фактори.** Рукотворні техногенні аварії (неправильне керування й обслуговуванням, пожежі, фізичний вплив, поломки, аварії, витоки).
- **Природні явища.** Стихійні лиха, шторми, урагани, повені.

У кожного об'єкта критичної інфраструктури є уразливі місця й критично важливі елементи.

– **Уразливі місця** – це ті вузли об'єкта, які можуть виявитися вузьким місцем у його захисті, переборюючи які, порушники можуть реалізувати свої мети. Також це ті вузли, які будуть піддані максимальному руйнуванню з боку природних явищ, внаслідок злочинних дій або халатного обігу.

– **Критичні елементи** – це ті частини об'єкта, вихід з ладу яких однозначно приводить до припинення функціонування всього об'єкта в цілому.

Для захисту уразливих місць потрібне оснащення об'єкта технічними засобами, а для захисту критичних елементів – постійний моніторинг їхньої працездатності. Ці завдання повинні вирішуватися з однаково високим пріоритетом, але для цього потрібні зовсім різні технічні засоби.

Рішення безпеки для нафтогазової галузі

Нафтогазові компанії працюють у висококонкурентному секторі, де руйнування інфраструктури й виробництва пов'язане з величезними економічними втратами. Для забезпечення ефективної й безперервної роботи при

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

дотриманні всіх правил безпеки, безпеки й охорони навколишнього середовища мережне відеоспостереження відіграє важливу роль у всьому ланцюжку поставок.

Рішення Axis для критично важливих об'єктів:

– Забезпечення ефективності роботи. Інтеграція відеоспостереження у виробничу систему дозволяє віддалено контролювати ефективність виробництва й візуально перевіряти функціонування систем і виконання процесів. Це дозволяє прогнозувати обслуговування навіть у важкодоступних місцях і надавати віддалене обслуговування із плановим обслуговуванням.

– Захист периметра нафто- і газо- трубопроводів. Трубопроводи є найбільш уразливою частиною ланцюжка поставок і самою дорогою частиною для захисту. Для ефективного захисту мережі трубопроводів від ушкоджень застосовують віддалений моніторинг.

Рішення Axis з захисту периметра трубопроводів:

– Гнучке керування інцидентами. Одним із пріоритетних напрямків є захисту співробітників.

– Інтеграція мережного відеоспостереження із системами HSE (охорони праці) допомагає забезпечити дотримання правил і процесів безпеки, а також правильне керування інструментами й устаткуванням.

Рішення Axis надають важливу інформацію в режимі реального часу для ідентифікації небезпеки, дозволяють негайно реагувати на інциденти.

Крок 2. Побудувати концепцію безпеки

Перед створенням системи безпеки для будь-якого об'єкта, у тому числі об'єкта із критичною інфраструктурою, необхідно визначити всі потенційні погрози, описати можливі небезпеки й ризики й створити моделі можливих порушників. На підставі цієї оцінки створюється концепція, що відповідає на питання: як у цілому повинен виглядати процес захисту даного об'єкта.

Також у концепції передбачаються кроки по ліквідації й мінімізації наслідків надзвичайних ситуацій, коли їх неможливо запобігти.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Таким чином, концепція безпеки об'єкта є **головним документом**, з якого починається побудова системи безпеки. На основі концепції згодом будуть створюватися технічні завдання та інші документи.

Після створення концепції стає очевидним, що комплексної інтегрованої системи – must have для об'єктів із критичною інфраструктурою.

Крок 3. Зробити вибір на користь інтегрованої системи

Ми є прихильниками комплексного підходу до побудови систем безпеки, оскільки це вигідно, дозволяє мати широкий набір інструментів і індивідуально його застосовувати залежно від завдань того або того або іншого об'єкта.

- Ефективність і економія.
- Інтегровані системи.

Ефективність і економія

У рамках великої системи комплексний підхід дозволяє одержувати більше повні дані від різних підсистем і використовувати їх більш ефективно. Наприклад, при виникненні надзвичайних ситуацій різного характеру комплексна система безпеки дозволить правильно маршрутизувати інформацію про цьому й передати її різним учасникам процесу. Таким чином, служби, що беруть участь у ліквідації наслідків НС, будуть одержувати однакову інформацію й зможуть діяти узгоджено.

Крім того, ті самі технічні засоби можуть працювати на різні служби, дозволяючи заощаджувати засоби, на відміну від ситуації, коли система складається з декількох дублюючих один одного технічних засобів, що належать різним організаціям і відомствам. При цьому концепція комплексного підходу сама по собі припускає індивідуальний набір рішень для кожного підрозділу, а значить і для різних уразливих місць об'єктів із критичною інфраструктурою.

Крок 4. Віддати перевагу системі, що використовує нейроаналітику

Зараз концепція інтегрованої системи стрімко розвивається у бік відеоаналізу. Якщо якийсь час назад відеоаналітика на інфраструктурних і критично важливих об'єктах використовувалася в основному для охорони

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

периметра в стерильних зонах разом з фізичними засобами виявлення, то зараз всі частіше знаходять застосування саме відеоаналітика реального часу, побудована на нейромережевих фільтрах.

Правильно навчені нейромережі дозволяють заздалегідь розпізнати погрози й максимально знизити можлива втрата. У той час як звичайне відеоспостереження й прості модулі відеоаналітики, начебто детектора руху, дозволяють відповістити на запитання «чи відбувається щось у кадрі», тобто зафіксувати тільки факт самої події, у системах машинного зору нейромережеві фільтри здатні аналізувати що відбувається й відповідати на запитання «що відбувається».

А саме головне, що з'являються інструменти, які дозволяють створювати кастомизовані унікальні нейрофільтри, орієнтовані на конкретні завдання конкретних підприємств.

При цьому також можливо одержання синергетичного ефекту від сполучення інформації, що генерує нейромережева аналітика, з інформацією від інших джерел. Це робить систему безпеки або детектування погроз на ранніх стадіях ще більш ефективною.

Нейромережеві фільтри дозволяють підняти систему безпеки на зовсім новий рівень, так само як у свій час інтеграція периметральної сигналізації з відеоспостереженням у стерильних зонах дозволила звести до нуля негативні наслідки помилкових спрацьовувань за рахунок того, що оператор одержав можливість миттєво візуально ідентифікувати джерело тривоги.

Крок 5. Вибрати правильну інтегруючу платформу

Неправильний вибір платформи може привести до того, що система безпеки не буде задовольняти поставленим завданням. Це може викликати серйозні наслідки, такі як необоротні негативні зміни в економіці й істотне погіршення безпеки життєдіяльності людей. Тому, ґрунтуючись на досвіді роботи в сфері безпеки, ми можемо із упевненістю сказати, що головний критерій вибору програмної платформи для побудови системи безпеки критично важливого

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

інфраструктурного об'єкта – це її максимально точна відповідність раніше створеної концепції безпеки.

При цьому бажано, щоб обрана програмна платформа була гнучкою й адаптивною, здатною без істотних фінансових витрат і людських ресурсів розширюватися разом з ростом самої системи і її елементів.

Енергетична компанія впроваджує відеоспостереження в інтересах майбутнього розвитку

Компанія **TAQA Energy B.V.** веде видобуток газу на родовищі Берхермеер на півночі Нідерландів починаючи з 1970-х років. Родовище вже майже вичерпане, але як і раніше являє комерційну цінність. Піщаник, у якому мільйони років залягав газ, тепер можна використовувати для підземного зберігання газу.

Рішення Axis для критично важливих об'єктів

Для забезпечення безпеки й контролю території була вибрана компанія **Spyke Security B.V.**, що зупинила свій вибір на системі відеоспостереження, повністю заснованої на камерах Axis.

Система містить у собі одну камеру з HD-дозволом і можливістю роботи в інфрачервоному діапазоні, що служить для виявлення сторонніх об'єктів – у будь-який час доби й у будь-яку погоду. Її доповнюють чотири тепловізійні камери.

Із червня 2010 року, коли були встановлені камери Axis, **TAQA Energy B.V.** безперебійно одержує потік відеоінформації, що дозволяє відслідковувати роботу інженерів і субпідрядників через захищене Інтернет-З'єднання. Зображення з камер у реальному часі допомагає управляти роботами на території й підтримувати безпека.

Компанія **Spyke Security** контролює зображення з камер винятково з метою безпеки.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Рішення Axis для критично важливих об'єктів

Основні завдання системи виявлення несанкціонованого проникнення

Всі елементи повністю оптимізованої охоронної системи повинні бездоганно взаємодіяти один з одним, відразу ж подаючи тривожні сигнали про будь-які порушення:

- Ідентифікація. Розпізнавання одушевлених і неживих об'єктів на конкретних ділянках великої території за допомогою функцій панорамування, нахилу й зума.

- Виявлення й спостереження. Захист території від небажаних дій, спостереження за тими, що рухаються, одушевленими й неживими об'єктами з реєстрацією пройденої відстані, кута й швидкості переміщення.

- Виявлення й перевірка. Застосування тепловізійних технологій для захисту периметра від несанкціонованого проникнення в режимі реального часу при слабкому освітленні або в повній темряві

- Контроль доступу. Спостереження за входами-виходами з наданням доступу тільки вповноваженому колу осіб

- Керування, що попереджає, тривожною сигналізацією. Централізоване або дистанційне керування охоронною системою за допомогою ПЗ для керування відеоспостереженням. Забезпечуючи надійний прийом тривожних сигналів і своєчасне на реагування на них, потужні інтелектуальні відеотехнології є гарантією безпеки охоронюваної території.

- Відлякування. Трансляція через рупорні гучномовці заздалегідь записаних голосових попереджень при тривожному сигналі про небажані дії.

Можливості для спостереження, захисти й патрулювання:

- Поліпшення контролю процесів. Відеозаписи Axis дозволяють виявляти відхилення у виробничому процесі – у тисках, потоках, температурах, витоках – і вчасно вживати заходів, щоб не допустити збою виробництва. Дистанційний візуальний контроль панелей керування й збір виробничих даних у

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

реальному часі дозволяють вчасно визначити, коли необхідне втручання. Результат – підвищення ефективності й зменшення простоїв.

– Охват більших площ. Захист і патрулювання протяжних периметрів у віддалених малолюдних місцях – нелегке завдання. Камери повинні бути здатні охоплювати значні площі, дозволяючи виявляти, локалізувати й розрізняти тварин, уповноважений персонал і порушників. Камери Axis з високошвидкісним механізмом PTZ (панорамування/нахил/ зум) дозволяють вирішити як це, так і інше завдання. Широкий огляд у сполученні з можливістю збільшувати будь-яку частину контрольованої зони підвищують рівень захищеності, забезпечують безпека й захищають ваше майно.

– Робота в темряві. Різке світло й темрява завжди ускладнюють захист майна й людей. Технологія Axis Lightfinder вирішує проблему, забезпечуючи необхідне вам чітке й детальне зображення. Напів тепловізійні камери здатні навіть у повній темряві автоматично виявляти підозрілі рухи й попереджати вас про їх.

– Зниження експлуатаційних витрат. Готовність до будь-яких ситуацій і здатність вчасно вживати заходів заощаджують ваші засоби в довгостроковій перспективі. Пропоновані Axis високоякісні відеокамери, тепловізійні камери, устаткування для роботи з аудіосигналом і даними допоможуть підвищити рівень захищеності вашого підприємства й позбутися від зайвих витрат на персонал. Завдяки відеокамерам ви зможете постійно контролювати територію підприємства в дистанційному режимі, оперативно виявляти неполадки й порушників і вживати необхідних заходів – направляти на місце аварійну бригаду або, наприклад, якщо вимагає ситуація, інформувати владу. Все це дозволяє прискорити ремонтні роботи й знизити збиток від позаштатних ситуацій.

Потужні засоби виявлення, охорони й патрулювання

Різні моделі камер AXIS і високотехнологічна система керування відео, інтегровані з існуючою системою безпеки й керування доступом, стають невід'ємною частиною системи керування інцидентами. У системі можливе

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

настроювання миттєвої передачі живого відео й відеозаписів у поліцію й інші органи.

Мережні камери AXIS серії Q60

Високошвидкісні купольні камери з 35-кратним оптичним зумом, що забезпечують видатну деталізацію в самих складних умовах відеоспостереження. Виняткове покриття більших територій.

Камери AXIS серії P13

Високоміцні фіксовані мережні камери із чудовою якістю відеозображення й дозволом до 5 мегапікселів. Ідеально підходять для спостереження за суспільними будинками, периметрами й т.д.

Тепловізійні камери AXIS серії Q19

Виявлення людей, об'єктів і подій у повній темряві й у складних умовах задимлення, мряки, пилу й туману. Ідеально підходять для цілодобового спостереження усередині й поза приміщеннями.

Камери AXIS серії P33

Вандало захищені фіксовані купольні відеокамери для будь-яких умов експлуатації з найвищими робочими характеристиками й дистанційним керуванням фокусуванням і зумом. Ідеальні для відеоспостереження в небезпечних зонах, таких як хімічні виробництва, промислові площадки й підприємства по переробці відходів.

Камери AXIS серії Q16

Винятково високі характеристики зображення в складних умовах і при погкій освітленості. Ідеальне рішення для установки в державних установах, на промислових об'єктах і підприємствах.

Системи відеоспостереження Axis дозволяють:

- забезпечувати безпека персоналу й охорону критично важливих об'єктів;
- контролювати ефективність виробництва;

– здійснювати спостереження, захист і патрулювання відповідальної інфраструктури.

КРОК. Захист критичної інформаційної інфраструктури

Послуги КРОК гарантують замовникам відповідність законодавству про безпеку критичної інформаційної інфраструктури (КІІ) і підзаконних актів. Впровадження цих рішень разом із проведенням ряду технічних-організаційно-технічних заходів дозволяє компаніям уникнути відповідальності за порушення законодавства й гарантувати безпеку об'єктів КІІ у своєму веденні.

1. Категоризування об'єктів КІІ

Фахівці КРОК визначають критичні процеси діяльності компанії, припинення яких може привести до негативних соціальних, економічних, екологічних наслідків, вплинути на безпеку держави й правопорядку.

Результат: замовник одержує перелік об'єктів, що підлягають категоризуванню відповідно до законодавства. Кожний з них одержує категорію значимості або обґрунтоване рішення про відсутність такої.

2. Розробка вимог до забезпечення безпеки

Кожна модель погроз містить короткий опис архітектури значимого об'єкта, характеристику джерел погроз безпеки інформації, у тому числі модель порушника, і опис самих погроз, актуальних для значимого об'єкта.

Результат: замовник одержує уточнені моделі погроз, які лягають в основу технічних завдань на системи безпеки.

3. Проектування систем безпеки

КРОК підготовляє технічні проекти й робочу документацію на системи безпеки значимих об'єктів КІІ.

Результат: опис архітектури систем безпеки по кожному об'єкті, опис порядку й параметри налаштування необхідних програмних і програмно-апаратних засобів.

4. Підготовка організаційно-розпорядницької документації

Експерти КРОК формують пакет документів, які регламентують обов'язку працівників, пов'язаних з експлуатацією, обслуговуванням і забезпеченням безпеки значимих об'єктів КІІ.

Результат: компанія має весь обсяг документів із правилами безпечної роботи співробітників, що працюють в об'єктах КІІ, у тому числі при позаштатних ситуаціях.

5. Впровадження систем безпеки

Установка й налаштування засобів захисту інформації, програмних і програмно-апаратних засобів, аналіз уразливостей значимого об'єкта й вживання заходів по їхньому усуненню.

6. Впровадження організаційних мір

Розробка правил, політики й стандартів для безпечної роботи з об'єктами КІІ.

Огляд, аналіз і рекомендації із захисту Критичної інформаційної інфраструктури (Critical Information Infrastructure, CII). Передовий досвід

Даний розділ ідентифікує передовий досвід захисту CII (Critical Information Infrastructure Protection, CIIP) серед держав-членів ЄС в області схем обміну інформацією між державними й громадськими організаціями й приватними особами, підготовки CII до надзвичайних ситуацій, зобов'язань і вимог до операторів CII.

Будуть надані приклади держав-учасників, які розробили рекомендується норми, що, у певній області. Список прикладів не є вичерпним і служить винятково як стимул для інших держав-членів ЄС.

Партнерство із приватними зацікавленими сторонами

Структурами ІСТ у різних галузях бізнесу в основному володіють приватні компанії, а це означає, що співробітництво державних організацій з операторами CII має критично важливе значення для того, щоб гарантувати, що

знання про поточні погрози є актуальними, і щоб забезпечити, при необхідності, швидке одержання підтримки у відповідь на інциденти.

Нідерланди являють гарний приклад надійного партнерства із приватним сектором. Національний центр кібербезпеки (NCSC, National Cyber Security Centre) служить як центральна точка для цілого ряду окремих-часткових-державно-приватних партнерств. У рамках NCSC було засновано кілька партнерств, основною метою яких стало виявлення, відгук і аналіз. Крім того, Рада по кібербезпеці (Cyber Security Council), що складається із представників державних організацій і приватних компаній, служить як консультативний орган. Завдяки цьому центр NSCS забезпечує тісне партнерство із приватними зацікавленими сторонами на стратегічному й операційному рівнях.

Ще одним прикладом тісного співробітництва із приватним сектором стала Австрія. Ця країна заснувала Платформу кібербезпеки (Cyber Security Platform), до складу якої входять представники приватних і державних операторів СП, а також відповідних державних агентств. Метою цієї платформи є полегшення комунікації між своїми учасниками. На операційному рівні була створена група GovCERT у якості головної державної CSIRT (групи реагування на інциденти, пов'язані з комп'ютерною безпекою). Нею керує Відомство федерального канцлера в співробітництві з CERT (приватна ініціатива).

Схеми обміну інформацією

Багато держав-члени ЄС розробили схеми обміну інформацією для того, щоб розподіляти важливу інформацію між відповідними державними агентствами й приватними операторами. Однак форми співробітництва можуть відрізнятися для різних країн. Схеми обміну інформацією гарантують, що всі релевантні зацікавлені сторони будуть проінформовані про поточні погрози й ризики й зможуть вжити відповідних заходів. Крім того, вони дозволяють зміцнити співробітництво й координацію дій і в такий спосіб сприяють ефективному використанню ресурсів.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Германія створила цілий ряд широкомасштабних схем обміну інформацією за участю державних і приватних агентств. Обмін інформацією між правоохоронними органами й спецслужбами реалізований через Національний центр кібербезпеки (National Cyber Response Centre)⁸, у рамках якого учасники інформують один одного в ході щоденних нарад і семінарів. Обміну інформацією із приватним сектором сприяють UP KRITIS і Alliance for Cyber Security: UP KRITIS⁹ – це державно-частка партнерство, його головним завданням є формування комунікацій і співробітництва в області СІПР між приватними й державними зацікавленими сторонами на стратегічному й операційному рівнях. У той час як UP KRITIS концентрує своя увага на співробітництві з компаніями критично важливих секторів, Alliance for Cyber Security (Альянс з кібербезпеки) має більше широкий порядок денний, у нього входять всі релевантні організації в області комп'ютерної безпеки. Для того, щоб зміцнити безпеку всіх зацікавлених сторін, альянс пропонує загальний «пул інформації», регулярні звіти про погрози й обмін знаннями між своїми учасниками (Federal Office for Information Security 2013, 2015).

Шведська Група співробітництва в області інформаційної безпеки (SAMFI, Cooperation Group for Information Security) – це ще один приклад передового досвіду в сфері обміну інформацією. На відміну від Німеччини, основна увага приділяється не правоохоронним органам і спецслужбам, а владі, відповідальним за безпеку соціальної й суспільної інформації. У рамках SAMFI різні органи влади не тільки обмінюються інформацією про недавні погрози, але й обговорюють стратегічні питання, а також національні й міжнародні тенденції розвитку. Представники різних органів влади зустрічаються кілька разів у рік і колективно працюють у складі робочих груп над рішенням поточних проблем. (Swedish Civil Contingencies Agency, MSB).

Розвиток співтовариства CSIRT

У більшості країн існують найрізноманітніші групи CSIRT з різними компетенціями. Утворення й розвиток міцного співтовариства за участю різних

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

національних груп CSIRT, включаючи поділ відповідальності й обмін інформацією, допоможе одержати взаємні переваги, наприклад, підвищення рівня знань і більше ефективне використання ресурсів.

Гарний приклад міцного співтовариства за участю груп CSIRT існує в Польщі. У Польщі жодна із груп CSIRT не одержала статус національної CSIRT. Замість цього співтовариство різних груп CSIRT спільно відповідає за безпеку. CERT.gov.PL являє собою головну групу CSIRT для державних установ, але вона також пропонує свої послуги операторам CI на основі формальних угод. CERT Polska стала першою групою CSIRT у Польщі, вона входить до складу Дослідницької й академічної комп'ютерної мережі (NASK, Research and Academic Computer Network). Ця група має спеціальні знання й досвідом в області аналізу й дослідження інцидентів інформаційної безпеки, а також надає інформацію про погрози й інциденти. Ця інформація доступна в базі даних, що можуть використовувати приватні й державні організації. CERT.gov.PL і CERT Polska працюють у тісному контакті, наприклад, управляючи базою даних, присвяченої мережам honeypot (незахищена мережа для вивчення прийомів хакерів). Більше того, вони співробітничать із цілим рядом секторальних груп CSIRT, таких як MilCERT і CERT Orange (телекомунікаційний сектор).

Іншими прикладами застосування передових наробітків при розвитку CSIRT-співтовариств є Нідерланди або Німеччина. У Німеччині був створений CERT-Verbund – альянс різних німецьких державних груп CSIRT (CERT-Bund, військова група CERTBw і кілька груп CERT федеральних земель), часток CSIRT найбільших компаній і груп CSIRT часток постачальників продуктів і послуг в області інформаційної безпеки (у числі інших учасників). Кожна група CSIRT як і раніше відповідає за власну аудиторію, однак учасники альянсу обмінюються інформацією й підтримують один одного при вирішенні інцидентів. CERT-Verbund – це інституціональний фундамент для такого співробітництва. Організація відкрита для прийому всіх типів груп CSIRT з Німеччини. Учасники визначили стандартизовані технічні й організаційні інтерфейси для обміну

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

інформацією. При цьому найважливішою частиною стає статистична оцінка спільно використовуваних даних і надання інформації стратегічного аналізу.

Оцінка ризику

Оцінка ризику включає ідентифікацію потенційних погроз, наслідку цих погроз (вплив) і їхня ймовірність. Аналіз ризиків є необхідним кроком підготовки й керування кризою й інцидентами. У деяких країнах національна оцінка ризику проводиться національним органом влади для всіх релевантних секторів. Інші країни, особливо ті з них, які використовують більше децентралізований підхід до СІПР, залишають оцінку ризику внутрішньогалузевим відомствам або операторам СІ.

Як приклад централізованої оцінки ризику на національному рівні можна привести Швецію. Шведський комітет MSB був уповноважений на продовження робіт з оцінки ризику на національному рівні, що почалися в 2011 році. Він розробив методологію оцінки ризику, ідентифікував 27 конкретних серйозних (національних) подій і розробив 11 сценаріїв на основі набору цих подій.

Данія не дотримується національного плану оцінки ризику, оскільки керування галузевими ризиками розглядається як більше успішний спосіб зм'якшення ризиків. При цьому був утворений Підрозділ по оцінці кіберзагроз (Cyber Threat Assessment Unit), до складу якого входять фахівці з різних галузевих органів державної влади. Метою цього підрозділу є проведення оцінок ризику для різних галузей.

Як приклад децентралізованого підходу можна привести Швейцарію. Швейцарія використовує підхід, що приділяє велику увагу індивідуальній відповідальності. Критично важливі підотраслі керують ідентифікацією ризиків інформаційної безпеки для своїх процесів і систем. Думають, що підотраслі найкраще знають свої власні процеси й системи. Держава підтримує цей процес при одержанні відповідного запиту.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Керування кризами інформаційної безпеки

Гарне керування кризами інформаційної безпеки включає визначення ролей і сфер відповідальності при надзвичайних ситуаціях в області інформаційної безпеки, а також координацію й процедури прийняття рішень за участю відповідних зацікавлених сторін, що володіють необхідними компетенціями й досвідом. Більше того, керування кризами інформаційної безпеки повинне бути погоджене з іншими існуючими національними системами антикризового й аварійного керування.¹²

Гарним прикладом передового досвіду в цій області є структура керування кризами інформаційної безпеки в Нідерландах. При прийнятті рішень застосовується Національна інструкція із прийняття рішень у кризових ситуаціях (National Manual on Decision-making in Crisis Situation) (National Coordinator for Security and Counterterrorism 2013). У випадку виникнення аварійної ситуації, пов'язаної з ІСТ, національна антикризова організація управляється директором по кібербезпеці Національного координатора по безпеці й антитерористичній діяльності (входить до складу міністерства безпеки й правосуддя). Заходу для оперативної взаємодії й антикризові міри пропонуються Національним центром кібербезпеки.

Керування кризами інформаційної безпеки здійснюється в тісному співробітництві із приватним сектором. У випадку виникнення кризи, пов'язаного з інформаційною безпекою, починає активну роботу Бюро оперативного реагування ІСТ. Це Бюро сформовано як державно-приватне партнерство, у його состав входять експерти по ІСТ з порушених галузей. Воно пропонує ради й рекомендації. Крім того, центр NCSC домовився з державними й приватними зацікавленими сторонами про методи антикризового керування.

Вичерпна правова основа

Деякі країни сформулювали проекти нових законів і нормативних актів для того, щоб вирішити проблему зростаючих погроз СІІ. Ця законодавча основа часто підганяється під потреби операторів СІ всіх релевантних галузей, а не

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

обмежується лише деякими з них. Ця законодавча основа часто включає обов'язкову реалізацію технічних і організаційних мір безпеки відповідно до національних і міжнародних стандартів. Інші вимоги можуть включати обов'язкове повідомлення про інциденти й регулярні зовнішні аудити систем безпеки. Ці закони гарантують однаковий рівень безпеки для всіх галузей і, у випадках обов'язкової звітності про стан безпеки, дозволяють державі аналізувати вхідні звіти про інциденти, відслідковувати рівні погроз, видавати попередження тим, що підверглися погрозі операторам СІ і, при необхідності, пропонувати підтримку.

Як приклад передового досвіду в цій області можна привести Францію. Закон про військове планування (LPM, Military Programming Law) зобов'язує «операторів особливої важливості» (OIV, operator of vital importance) повідомляти про інциденти кібербезпеки агентству Agence nationale de la sécurité des systèmes d'information (ANSSI), а також реалізовувати технічні й організаційні міри в області інформаційної безпеки. Крім того, OIV зобов'язані проводити аудити кібербезпеки, які повинні виконуватися або агентством ANSSI, або провайдером послуг, атестованим ANSSI (French Senate 2013). Аналогічні зобов'язання були прийняті в Німеччині, у якій недавно набув чинності Закон про ІТ-безпеку (IT Security Act)¹⁴. Недавно прийнятий закон зобов'язує операторів реалізовувати адекватні організаційні й технічні міри в області інформаційної безпеки в тім ступені, у якій це необхідно для забезпечення доступності їх критично важливих послуг. Більше того, організації повинні проводити аудитів безпеки, створювати контактні точки усередині своєї структури й повідомляти про істотні інциденти в області ІТ-безпеки, якщо була ймовірність, що вони могли торкнутися доступність критично важливих послуг.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як С.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на С або С++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.

Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з'являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп'ютері. Одна з основних функцій процесора – це обробка даних згідно комп'ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп'ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп'ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.
- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.
- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.
- Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.
- Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.
- PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи захисту мережевої IT-інфраструктури на об'єктах критичного застосування.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Технологічні зміни вже не просто константа, а прискорення. У 2026 році галузь виходить за рамки простого впровадження нових інструментів і тепер зосереджена на створенні стійких, інтелектуальних основ, необхідних для їх підтримки. У цьому розділі визначили ключові тенденції, які змінюють ландшафт цього року, від математичної основи кібербезпеки до фізичних обмежень потужності центрів обробки даних.

Кібербезпека

Кібербезпека у 2026 році – це не стільки зведення вищих стін, скільки підготовка до двох нових реальностей: ризиків квантової ери та світу, де ідентичність є периметром.

1. Гнучкість постквантової криптографії (PQC)

Постквантова криптографія перейшла від академічної теорії до активного впровадження. Основною загрозою, що керує цим зрушенням, є принцип «Збирай зараз, дешифруй пізніше» (HNDL), коли зловмисники збирають зашифрований трафік сьогодні з наміром розшифрувати його, як тільки квантові можливості стануть зрілими. Розумним кроком у 2026 році є ставлення до шифрування як до проблеми інвентаризації – відображення ваших найчутливіших даних та розвиток «гнучкості» для заміни криптографічних стандартів без руйнування критично важливих систем.

2. Нульова довіра, що ґрунтується на ідентичності.

Традиційний мережевий периметр мертвий; користувачі, програми та дані тепер постійно переміщуються між середовищами. У 2026 році ідентифікація стала єдиним периметром, який має значення. Безпека переходить до контекстно-залежного доступу, де довіра надається на основі сигналів у режимі реального

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

часу – хто користувач, що він робить та стан його пристрою – а не лише пароля. Такий підхід зменшує «радіус вибуху» інцидентів безпеки, гарантуючи, що викрадені облікові дані не перетворяться на кризу всього підприємства.

Архітектура центру обробки даних

Стратегія центрів обробки даних переписується фізичними реаліями штучного інтелекту, де обмежувальними факторами вже є не амбіції, а тепло та електроенергія.

3. Рідинне охолодження стає мейнстрімом (пряме охолодження на кристал).

Рідинне охолодження більше не є нішевим рішенням; воно стало стандартним рішенням для центрів обробки даних, готових до використання штучного інтелекту. Стандартне повітряне охолодження просто не може впоратися з щільністю кіловат на стійку найновіших кластерів графічних процесорів, які значно нагріваються, ніж традиційні процесори. Пряме охолодження на кристал тепер є стандартним конструктивним рішенням для підтримки стабільної продуктивності, необхідної для масштабних робочих навантажень штучного інтелекту.

4. Дефіцит електроенергії та локальні мікромережі.

Наявність електроенергії замінила землю як основне обмеження для розширення центрів обробки даних. Оскільки робочі навантаження штучного інтелекту не терплять перебоїв, оператори центрів обробки даних перетворюються на енергетичних стратегів. Ми спостерігаємо значне зростання локальних мікромереж та систем накопичення енергії для стабілізації постачання, захисту часу безперебійної роботи та зменшення залежності від напруженої громадської мережі.

Хмара

У 2026 році хмара вже не є єдиним місцем призначення; це стратегічне портфельне рішення щодо того, де кожне робоче навантаження належить і чому.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

5. «Велика репатріація» та розумний гібрид.

Після років імпульсу, спрямованого на хмарні технології, багато організацій уважно розглядають стаціонарні робочі навантаження та запитують себе, чи повинні вони володіти тим, що вони зараз орендують. Ця «хмарна репатріація» зумовлена контролем витрат; коли робочі навантаження є передбачуваними та тривалими, їх перенесення назад у локальні середовища або в колокейшн може відновити значний вплив на продуктивність та ціноутворення.

6. Зростання неохмар.

Злет обчислень на основі штучного інтелекту призвів до появи спеціалізованих «неохмар» – провайдерів, створених спеціально для високопродуктивних обчислень на графічних процесорах, а не для широких універсальних послуг. Ці платформи пропонують масштабованість та чіткішу економіку, необхідні для навчання, точного налаштування та високопродуктивного виведення ШІ, диверсифікуючи хмарну екосистему за межі традиційних гіпермасштабованих систем.

Мережеві рішення

Сучасні мережі стали надто складними для підходу «змінюйся та молись». Переможцями у 2026 році стануть команди, які симулюють свої дії, перш ніж впевнено їх розгортати.

7. Цифрові двійники мережі

Організації все частіше впроваджують цифрових двійників мережі – живі, віртуальні репліки топології своєї інфраструктури, політик та моделей трафіку. Інженери тепер можуть моделювати вплив нового правила сегментації або зміни політики брандмауера в безпечному середовищі, перш ніж воно потрапить у виробництво. Це перетворює оновлення мережі на передбачуваний, перевірений процес.

8. Wi-Fi 7 як стандарт оновлення

Wi-Fi 7 тепер є стандартом за замовчуванням для всіх нових оновлень мережі. Хоча швидкість є головним пріоритетом, справжня бізнес-цінність

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

полягає в значно нижчій затримці та кращій надійності, необхідних для сучасних робочих місць, де використовується співпраця в режимі реального часу, робочі процеси на основі штучного інтелекту та захопливе навчання.

Штучний інтелект

Штучний інтелект – це вже не просто швидша генерація тексту; це виконання ШІ реальної роботи безпечно, повторювано та в масштабі підприємства.

9. Агентний ШІ переходить у виробництво.

Ми перейшли від генеративного ШІ (який пише) до агентного ШІ (який діє). Ці автономні цифрові оператори можуть інтерпретувати цілі, розбивати їх на кроки та викликати інструменти для виконання завдань з чітким журналом аудиту. Впровадження спочатку торкається виробництва в ІТ-операціях, запитах на закупівлі та робочих процесах сприяння продажам.

10. Вплив штучного інтелекту на запобігання втраті даних (DLP).

Штучний інтелект переписує принципи запобігання втраті даних, щоб керувати взаємодією зі штучним інтелектом, а не просто блокувати передачу файлів. Ризик випадкового вставки співробітниками конфіденційної інформації компанії в публічні чат-боти є критично важливим. Сучасні рішення тепер застосовують політики щодо запитів та завантажень, спрямовуючи користувачів до безпечних, схвалених помічників на базі штучного інтелекту.

11. Інтелектуальний край

Інтелектуальний край забезпечує обробку даних у режимі реального часу та прийняття рішень на основі штучного інтелекту ближче до джерела даних – чи то на заводі, в лікарні чи роздрібному магазині. Такий підхід значно зменшує витрати на затримку та пропускну здатність, водночас забезпечуючи стійкість, коли підключення до централізованої хмари обмежене.

ІТ-закупівлі

Закупівлі у 2026 році – це вже не просто «купівля інструментів»; це управління ризиками та постійне підтвердження результатів.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

12. Моніторинг ризиків постачальників у режимі реального часу.

Управління ризиками постачальників перейшло від щорічних контрольних списків до «динамічної» трансляції. Інструменти штучного інтелекту тепер відстежують стан постачальників – відстежуючи такі сигнали, як інциденти безпеки, фінансові труднощі та нестабільність ланцюга поставок – у режимі реального часу, щоб прогнозувати та запобігати збоям до їх виникнення.

13. Команди з впровадження технологій, орієнтовані на результат,

все частіше купують вимірювані результати, а не окремі компоненти. Ця методологія зосереджена на узгодженні інвестицій у технології з чіткими бізнес-цілями, такими як скорочення часу на вирішення проблеми або зниження вартості інфраструктури на кожне робоче навантаження. Це гарантує, що витрати безпосередньо пов'язані зі стратегією – зрештою, якщо результат не визначено, покупка – це лише надія з бюджетом.

Підготовка до майбутнього

ІТ-інфраструктура 2026 року визначається поєднанням готовності, модернізації та стратегічного розміщення. Стрибок у ці переходи без підготовки – це як спроба побудувати будинок на піску; результати навряд чи будуть стабільними чи сталими. Справжній успіх починається з чіткого розуміння якості ваших даних, можливостей інфраструктури та конкретних проблем, які ви прагнете вирішити.

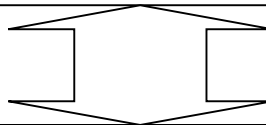
3.2 Розробка структурної схеми

Структурна схема містить у собі наступні основні блоки:

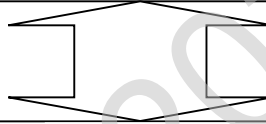
- Блок категоризації критичної інформаційної інфраструктури.
- Блок SIEM (Security information and event management).

Блок категоризації критичної інформаційної інфраструктури

- Виявлення об'єктів критичної інформаційної інфраструктури, які забезпечують управлінські, технологічні, виробничі, фінансово-економічні й (або) інші процеси у рамках діяльності суб'єкта критичної інформаційної інфраструктури
- Виявлення критичних процесів, порушення або припинення яких може привести до негативних наслідків у соціальній, політичній, економічній і оборонній сферах
- Встановлення об'єкти критичної інформаційної інфраструктури, які обробляють інформацію для описаних процесів і/або здійснюють керування, контроль або моніторинг критичних процесів
- Побудова моделі порушників і погроз
- Оцінка можливих наслідків
- Привласнення кожному об'єкту одну із трьох категорій значимості або винесення мотивованого рішення про неприєднання такої категорії



Система захисту мережевої IT-інфраструктури на об'єктах критичного застосування



Блок SIEM

- Інвентаризація інформаційних ресурсів
- Виявлення уразливостей інформаційних ресурсів
- Аналіз погроз інформаційної безпеки
- Підвищення кваліфікації персоналу інформаційних ресурсів
- Прийом повідомлень про можливі інциденти від персоналу й користувачів інформаційних ресурсів
- Виявлення комп'ютерних атак
- Аналіз даних про події безпеки
- Реєстрація інцидентів
- Реагування на інциденти й ліквідація їхніх наслідків
- Установлення причин інцидентів
- Аналіз результатів усунення наслідків інцидентів

Рисунок 3.1 – Структурна схема системи

Отже, суб'єкт критичної інформаційної інфраструктури, що ставиться до органа державної влади, відповідно до поточних законодавчих норм повинен підключитися до відповідного відомчого Центра SIEM. У суб'єкта критичної інформаційної інфраструктури, що не є органом державної влади, є можливість

або здійснити самостійне підключення до системи SIEM, або створити свій власний корпоративний Центр SIEM, або підключитися до вже створеного Центра, що робить послуги з підключення до системи SIEM.

Підключення до зовнішнього (комерційному) центру SIEM, що робить відповідні послуги, дозволяє передати більшість вищеописаних завдань спеціалізованої організації. Від замовника буде потрібно лише підключити джерела подій до комерційного центра SIEM, погодити формат і регламент взаємодії, а також вчасно повідомляти про зміни у своїй IT-інфраструктурі. Крім того, скориставшись послугами зовнішнього центра SIEM, організація перекладає на виконавця ризики невідповідності законодавству України в області ведення незаконного підприємництва.

Штучний інтелект, SecDataOps та федерація – це одні з основних сил, які продовжуватимуть змінювати операції з безпеки в найближчі роки. Чого саме нам очікувати у 2026 році? Ми все ще в січні, тому я хотів би втиснути свої прогнози на 2026 рік у галузі моєї експертизи!

Дивлячись уперед на 2026 рік, стратегії безпеки даних – і роль SIEM – готові до значних змін. Спосіб, у який команди безпеки збирають, зберігають та досліджують дані, вже кілька років перебуває на нестійкому шляху за традиційних парадигм SIEM. Зростання витрат, розростання даних, численні хмарні облікові записи та регіони, а також перевантаження аналітиками виявили фундаментальні обмеження централізованих платформ SIEM.

Нижче наведено п'ять головних прогнозів, які, на нашу думку, визначатимуть дані безпеки та стратегію SIEM у 2026 році.

1. Завдяки конвеєрній розробці нового покоління, хмарні сховища та озера даних зростатимуть більше, ніж SIEM.

Не всі дані належать до SIEM – і не повинні. Централізація історичних журналів, довгохвостової телеметрії та джерел великого обсягу часто є дорогою в SIEM. Хмарні озера даних та сховища BLOB-об'єктів стануть заміною точки збору. Але для цього спочатку потрібно виправити конвеєрну роботу.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Для конвеєрної обробки даних новим критерієм буде вибіркове, надійне, оптимальне та продуктивне посівне навантаження. Застарілі реалізації конвеєрів залишаються складними, крихкими та складними проектами для налаштування та запуску. Архітектура безпеки перейде до більш вибіркової та цілеспрямованої стратегії конвеєрної обробки даних у 2026 році. У цій моделі:

- Сповіщення безпеки продовжують надсилатися до вашої SIEM-системи для аналізу в режимі реального часу.
- Об'ємні журнали надходять до економічно ефективних хмарних сховищ та озер даних. Дані з довгим хвостом та дані про відповідність зберігатимуться в озерах з можливістю пошуку або сховищах об'єктів.
- Будуть впроваджені рішення для конвеєрної реалізації, які є надійними, простими в налаштуванні та управлінні (без необхідності інженерії) та створюють оптимальний золотий шар даних.
- Федеративний пошук за потреби здійснюватиме перехід між місцями зберігання.

Ця стратегія відповідає менталітету мережі даних, де кожен тип даних зберігається, структурується та запитується на основі його реальної бізнес-корисності.

2. «SIEM як лінза», а не система обліку, стане частиною стратегій міграції SIEM.

Традиційна SIEM-система здавна базувалася на централізації всіх даних безпеки в одному сховищі, але ця модель руйнується під власною вагою:

- Стрімке зростання витрат на споживання та зберігання.
- Ізоляції даних, що охоплюють SaaS, хмарні облікові записи, мережеву телеметрію та застарілі системи.
- Команди аналітиків змушені приховувати джерела, щоб контролювати обсяг.

У 2026 році консолі SIEM стануть аналітичними механізмами, а не сховищами даних. Важка робота зі зберігання та утримання даних переміститься

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

до спеціально створених сховищ – хмарних об'єктних сховищ, озер та нативних сховищ, розміщених постачальниками, – тоді як новий SIEM зосередиться на виявленні та візуалізації цих децентралізованих даних.

Таке розмежування консолі, збору та зберігання відображає те, як переосмислюються сучасні архітектури даних безпеки. Див. статтю «Міграція SIEM», щоб дізнатися більше про те, як відбуватиметься перехід від застарілих SIEM у 2026 році.

3. Федеративний пошук та мережа даних погіршать централізацію як основний шаблон

Часи, коли навантажувачі повторно завантажували дані в монолітну SIEM-систему, минули. Команди SecOps хочуть мати доступ до місця зберігання даних – у кількох хмарних облікових записах та регіонах, на платформах SaaS постачальників, у застарілих сховищах журналів та сучасних озерах даних – без дублювання.

У 2026 році федеративний пошук стане основоположним стандартом для запитів даних безпеки. Аналітики більше не будуть думати лише «спочатку отримати, потім запитувати» – вони думатимуть «шукати скрізь тепер».

Цей перехід дозволяє:

- Доступ до високоякісних джерел у режимі реального часу.
- Стратегії зберігання з контрольованими витратами.
- Уніфікований контекст розслідування без дублікатів сховища.

Організації, які впроваджують федеративні архітектури (тобто мережу даних безпеки), значно зменшать накладні витрати, зберігаючи при цьому повноваження щодо розслідувань. Більше про ці аспекти дивіться в моєму попередньому блозі про п'ять модифікацій для уявлення про нову архітектуру SIEM.

4. Штучний інтелект з федеративним інтелектом трансформує виявлення та розслідування.

Сучасні SIEM-системи спираються на правила та пакети контенту, які

припускають, що всі дані вже централізовані. Але з огляду на те, що дані розпорошені по різних системах та форматах, отримання змістовної аналітики стало складнішим, а не легшим.

У 2026 році штучний інтелект стане функцією примусового виконання єдиного рівня, який інтерпретуватиме, співвідноситиме та контекстуалізуватиме дані безпеки в розподілених середовищах.

Замість ручного зшивання запитів, аналітики використовуватимуть логіку федеративного виявлення на основі штучного інтелекту для:

- Автоматично генерувати кросплатформні запити на виявлення загроз;
- Співвідносити гетерогенні схеми та моделі даних – як засоби забезпечення ШІ знадобляться Security Data Mesh та OCSF;
- Виявляйте поверхневі аналітичні дані, які не можна було б спостерігати, якби ваша SIEM-система не обробила ці дані.

Цей стрибок відображає ширші галузеві тенденції, де штучний інтелект використовується для пошуку сенсу в даних.

5. Впровадження штучного інтелекту не обмежується лише MCP та агентним підходом. Загальні галузеві моделі даних, такі як OCSF, стануть ключовими для розуміння ШІ даних безпеки.

Оскільки штучний інтелект почне вбудовуватися в операції безпеки, організації виявлять гірку правду: ШІ настільки хороший, наскільки хороші дані, які він споживає. Простого надання інструментів через MCP, A2A та інші агентні платформи недостатньо. Окрім обмежень токенів для величезних обсягів даних, організації також усвідомлять, що дані є безладними, фрагментованими та семантично несумісними.

У 2026 році Open Cybersecurity Schema Framework (OCSF) надасть перевагу організаціям, які прагнуть досягти результатів безпеки на основі штучного інтелекту. Чому? Тому що системам ШІ потрібно:

- Нормалізований та локалізований контекст у хмарних, SaaS, ідентифікаційних, кінцевих та мережевих даних.

- Послідовне семантичне значення в різних джерелах даних з передбачуваною схемою для міркування, кореляції та навчання.
- Зменште припущення та галюцинації.

Без стандартизованих моделей даних ШІ змушений постійно навчатися або припускати структуру, замість того, щоб зосереджуватися на аналітиці. OCSF зміщує проблему вліво – дозволяючи ШІ міркувати над уже змодельованими зв'язками між подіями та сутностями, що містять відповідні ІТ-об'єкти та об'єкти безпеки. Це являє собою фундаментальну еволюцію, тобто від оптимізації для аналітиків-людей до оптимізації для виявлення, розслідування та реагування за допомогою ШІ.

Окремі постачальники рішень безпеки запізняються з підтримкою OCSF, тому цього не станеться у 2026 році. Але ви можете використовувати Query Security Data Mesh, оскільки це шлюз до даних будь-якого постачальника, перетворених в OCSF.

Майбутнє SIEM та штучного інтелекту в галузі безпеки полягає у впровадженні розподілених даних з рівнем федерації.

Майбутнє SIEM визначається не потужнішими механізмами обробки даних чи більшою централізацією. Воно залежить від того, наскільки добре дані безпеки структуровані, доступні та зрозумілі – як людьми, так і машинами.

У 2026 році команди безпеки-переможці отримають:

- SIEM як рівень аналітики, а не сховище даних.
- Федеративний пошук замість примусової централізації.
- Штучний інтелект як власний учасник розслідувань.
- Цільові канали передачі даних безпеки.
- Стандарти відкритих даних, такі як OCSF, роблять штучний інтелект справді ефективним.

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. Функціональна схема розроблена для прикладу передачі інформації у системі ІТ-інфраструктури на об'єктах критичного застосування. Рішення задачі забезпечення своєчасності передачі інформації в мережах систем ІТ-інфраструктури на об'єктах критичного застосування складається з чотирьох етапів:

- визначення множини $\aleph_{\text{баз}}$ маршрутів передачі інформації;
- знаходження оптимальної множини маршрутів передачі цифрової інформації в телекомунікаційній мережі $\aleph_{\text{об}}$;
- обчислення коефіцієнтів $\tilde{k}_s$ розподілу інформаційного потоку і управління навантаженням комп'ютерної мережі систем критичного застосування;
- створення та оновлення таблиці маршрутизації.

При рішенні задачі визначення множини $\aleph_{\text{баз}}$ шляхів передачі інформації в мережах систем ІТ-інфраструктури на об'єктах критичного застосування для ВЗ «і» та «j» з множини $\aleph$ вузлів зв'язку спочатку необхідно знайти найкоротшу «відстань» (мінімальний час передачі інформаційних пакетів) $T_{i,j \min}$ від джерела «і» до адресата «j» і множини $S_j^{(i)}$ вузлів, найближчих ВЗ «і» за напрямом руху потоку до «j» (множина «вузлів-наступників») у порядку рівнів ієрархії дерева допустимих маршрутів множини U . При рішенні поставленої в (3.11)–(3.15) задачі відомими алгоритмами пошуку найкоротших шляхів в більшості практичних випадків маємо проблему «зациклення» при передачі інформації в знайдених шляхах. Це призводить до збільшення часу передачі інформаційних пакетів, а деколи і до їх втрати. Уникнути «зациклення» при передачі інформації пропонується шляхом додання обмежень (умова постійної відсутності циклів), які надані у вигляді виразів:

$$T_{k,j} \leq T_{i,j \min}; \quad (3.1)$$

$$T_{k,j \min} \leq T_{i,k,j}, \quad k \in R, \quad (3.2)$$

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

де $T_{k,j \min}$ – найкоротша «відстань» (мінімальний час передачі інформаційних пакетів) від вузла « k » до адресата « j »; $T_{i,k,j}$ – «відстань» (час передачі інформаційних пакетів) від вузла « i » до адресата « j » через вузол « k ».



Рисунок 32 – Функціональна схема системи

Рішення задачі пошуку множини шляхів, що виключають «цикли», складається з двох етапів: визначення найкоротшої «відстані» $T_{i,j \min}$ від джерела « i » до адресата « j »; знаходження множини $S_j^{(i)}$ «вузлів-наступників» на маршрутах, що виключають «циклічність», для довільних джерел « i » та адресатів « j » за порядком множини U рівнів ієрархії дерева вибору допустимих маршрутів.

На першому етапі для визначення найкоротшої «відстані» $T_{i,j \min}$ доцільно використати алгоритм розрахунку попередньої топології, який забезпечує вузли зв'язку інформацією про стан зв'язків для обчислення найкоротших шляхів до адресатів. Алгоритм створено на основі відомих алгоритмів стану зв'язків. На відміну від відомих, в цьому алгоритмі враховується ієрархічність побудови комп'ютерної мережі систем критичного застосування (визначаються «відстані» від «вузла-джерела» i до «адресата» j відповідно до існуючих рівнів ієрархії), що надалі дозволить здійснити пропорційний (з урахуванням коефіцієнтів k_s і $k_s^{(c)}$) розподіл інформації по знайдених маршрутах.

На другому етапі для знаходження множини шляхів передачі інформації, що виключають «циклічність», використовується алгоритм розповсюдження попередньої топології множини шляхів. На відміну від відомих алгоритмів, в яких не враховується поточний стан ВЗ « i » обов'язкова синхронізація обміну службовими повідомленнями про стан зв'язків по всій мережі, в алгоритмі розповсюдження попередньої топології множини шляхів такої синхронізації підлягає тільки один перехід між сусідніми вузлами. Це значно спрощує роботу вузлів зв'язку і скорочує час збіжності алгоритмів. Сукупність алгоритмів розрахунку попередньої топології і розповсюдження попередньої топології множини шляхів є основою способу визначення множини шляхів передачі інформації.

Запропонований спосіб за відсутністю флуктуацій трафіку зіставлений з аналогічними за часом затримки передачі інформації, проте у декілька разів перевершує їх при різких флуктуаціях вхідного трафіку.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Безпосереднє використання всієї знайденої множини $\aleph_{\delta az}$ шляхів передачі інформації розробленим способом не завжди є виправданим, особливо у разі високої пропускної спроможності декількох з наявних каналів зв'язку, здатних забезпечити виконання вимог:

$$T_{\text{срд}}(\bigcup_{u_{\alpha} \leq u_{\alpha}} \aleph(u'_{\alpha})) \leq T_{\text{дон}} \quad \forall u_{\alpha} \in U; \quad (3.3)$$

$$Q^{(cy)}(\bigcup_{u_{\alpha} \leq u_{\alpha}} \aleph(u'_{\alpha})) \geq Q_{\text{дон}}. \quad (3.4)$$

при передачі інформації про повітряну обстановку. Розширення такої множини призводить до збільшення таблиць маршрутизації вузлів зв'язку, ускладнення процесу розподілу інформації і, як наслідок, до зниження достовірності передачі цифрової інформації. Тому виникає необхідність в знаходженні такої топології підмережі, тобто у виборі зі всієї знайденої множини $\aleph_{\delta az}$ шляхів деякої (оптимальної) сукупності $\aleph_{\delta \theta}$ маршрутів, використання якої в умовах обмежень, що накладаються, дозволить забезпечити максимально можливу достовірність передачі інформації.

З урахуванням:

$$q(\aleph_{\delta az}, \Delta t) = 1 - \prod_{s \in \aleph_{\delta az}} \prod_{c \in \eta_s} (1 - q_s^{(c)})^{\lambda_{k_s} \lambda_p \Delta t}. \quad (3.5)$$

в умовах, описаних вище, одержані і наведені на рисунку 3.6 криві залежності ймовірності $q(\aleph_{\delta az}, \Delta t)$ від λ . Збільшення числа M використаних маршрутів, призводить до істотного (у 2,3...3,4 разів) збільшення ймовірності $q(\aleph_{\delta az}, \Delta t)$ спотворення інформації в процесі її передачі. Слід особливо відзначити, що для відомих методів розподілу характерна відсутність моніторингу поточного завантаження маршрутів, змін вхідного потоку інформації, а інколи і технічного стану каналів зв'язку. Принцип рівномірного завантаження вибраних M маршрутів, який використовується в таких методах, призводить до перевантаження одних і до неефективного використання інших маршрутів.

Для визначення початкового завантаження підмережі комп'ютерної мережі систем критичного застосування пропонується проводити розрахунок коефіцієнтів

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

$k_s^{i,k,j}$ розподілу потоку інформації від джерела « i » до адресата « j » між «вузлами-наступниками» « k » з M -мірної множини $S_j^{(i)}$ залежно від значень «відстані» (часу передачі пакетів) між ВЗ на маршруті за співвідношенням

$$k_s^{i,k,j} = \left(1 - \left(T_{k,j} + t_{i,k} \right) / \sum_{\xi \in S_j^{(i)}} (T_{\xi,j} + t_{i,\xi}) \right) / \left(|S_j^{(i)}| - 1 \right), \quad k \in S_j^{(i)}, \quad (3.6)$$

де $T_{k,j}$ – час передачі пакетів від «вузла-наступника» « k » до адресата « j » на маршруті (i, j); $t_{i,k}$ – «відстань» від джерела « i » до «вузла-наступника» « k »; $|S_j^{(i)}|$ – потужність множини $S_j^{(i)}$.

Проведені заміри залежності середнього часу $T_{ср\partial}$ доставки інформаційних пакетів в мережах систем ІТ-інфраструктури на об'єктах критичного застосування і ймовірність $Q^{(сч)}$ їхньої доставки за час, що не перевищує допустиме значення, від інтенсивності λ вхідного потоку інформації для повнозв'язного фрагменту комп'ютерної мережі систем критичного застосування в умовах мінімальної ($\rho_{min} = 16$ Кбіт/с), максимальної ($\rho_{max} = 30$ Кбіт/с), середньої ($\rho_z = 19$ Кбіт/с) пропускної спроможності каналів зв'язку (умови (І) – суцільні криві), $\rho_{min} = 14$ Кбіт/с, $\rho_{max} = 300$ Кбіт/с, $\rho_z = 70$ Кбіт/с (умови (ІІ) – пунктирні криві) при кількості маршрутів $M=5$. Параметром сімейства кривих є метод розподілу мережевого ресурсу в мережах систем ІТ-інфраструктури на об'єктах критичного застосування.

Аналіз залежностей показав, що у деяких ситуаціях адаптивний розподіл потоку інформації дозволяє «заощадити» 2 і більш маршрутів і тим самим додатково понизити ймовірність $q(N_{\partial аз}, \Delta t)$ порівняно з відомими методами управління, що свідчить про необхідність оптимізації топології підмережі комп'ютерної мережі систем критичного застосування. В роботі пропонується метод знаходження оптимальної множини маршрутів в мережах систем ІТ-інфраструктури на об'єктах критичного застосування, який включає постановку завдання та процедуру знаходження оптимальної множини маршрутів в мережах систем ІТ-інфраструктури

на об'єктах критичного застосування. При цьому визначення початкового завантаження підмережі дозволяє найкращим чином розподілити потік пакетів, тим самим мінімізуючи $T_{ср\partial}$ і максимізуючи $Q^{(сч)}$ в даних умовах.

В подальшому, для передачі інформації про повітряну обстановку та забезпечення «збалансування» загрузки при флуктуаціях трафіку виконується процедура розподілу інформаційних пакетів за оптимальною множиною маршрутів в мережах систем ІТ-інфраструктури на об'єктах критичного застосування.

Проведемо оцінки ефективності методу забезпечення своєчасності передачі інформації в мережах систем ІТ-інфраструктури на об'єктах критичного застосування при передачі інформації про повітряну обстановку і вірогідності отриманих результатів. В якості рекомендацій щодо практичного застосування методу забезпечення своєчасності передачі інформації в мережах систем ІТ-інфраструктури на об'єктах критичного застосування при управлінні повітряним рухом розроблена структура процесу передачі інформації про повітряну обстановку в режимі реального часу. Врахована необхідність здійснювати комплекс заходів (адаптивне кодування, зменшення статистичної і психовізуальної надмірності, адаптивна маршрутизація та ін.), направлених на управління швидкістю передачі інформації про повітряну обстановку і зниження її інтенсивності. Для компенсації втрачених при передачі в каналах зв'язку інформаційних пакетів і забезпечення безперервності відтворення інформації про повітряну обстановку розроблено алгоритм компенсації втрачених інформаційних пакетів. Це дозволить використовувати для управління передачею цифрової інформації про повітряну обстановку протоколи транспортного рівня (RTP і UDP).

Оцінка ефективності роботи методу забезпечення своєчасності передачі інформації в мережах систем ІТ-інфраструктури на об'єктах критичного застосування по відношенню до відомих методів показала ряд його переваг (Зймовірність доведення інформації до одержувача за час, що не перевищує допустиме значення, вище за аналогічну ймовірність відомих методів до 3 разів, середній час доставки інформаційних пакетів менше аналогічної характеристики відомих методів до 15

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

разів).

Для обґрунтування достовірності отриманих результатів математичного моделювання проведено імітаційне моделювання передачі інформації в повнозв'язній комп'ютерній мережі в умовах застосування бездротових каналів зв'язку (середня пропускна спроможність $\rho_z = 19 \text{ Кбіт/с}$) при різних інтенсивностях вхідного потоку (у діапазоні $\lambda = [5, \dots, 30] \text{ Кбіт/с}$).

За критерієм згоди Персона з рівнем значущості $\alpha = 0,01$ показано, що число прийнятих за час $T_{\text{дон}}$ інформаційних пакетів а також ймовірність $Q_{\text{експ}}^{(сч)}$ можна вважати розподіленими за нормальним законом.

Одержані довірчі інтервали для оцінок математичного очікування $Q_{\text{експ}}^{(сч)}$, в які потрапляють «розрахункові» значення $Q_{\lambda}^{(сч)}$ з довірчою ймовірністю 0,95. Високий ступінь збігу результатів імітаційного і математичного моделювання підтверджує достовірність математичної моделі, використаної для розрахунку ймовірності $Q^{(сч)}$ доставки інформаційних пакетів за час, що не перевищує допустиме значення.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

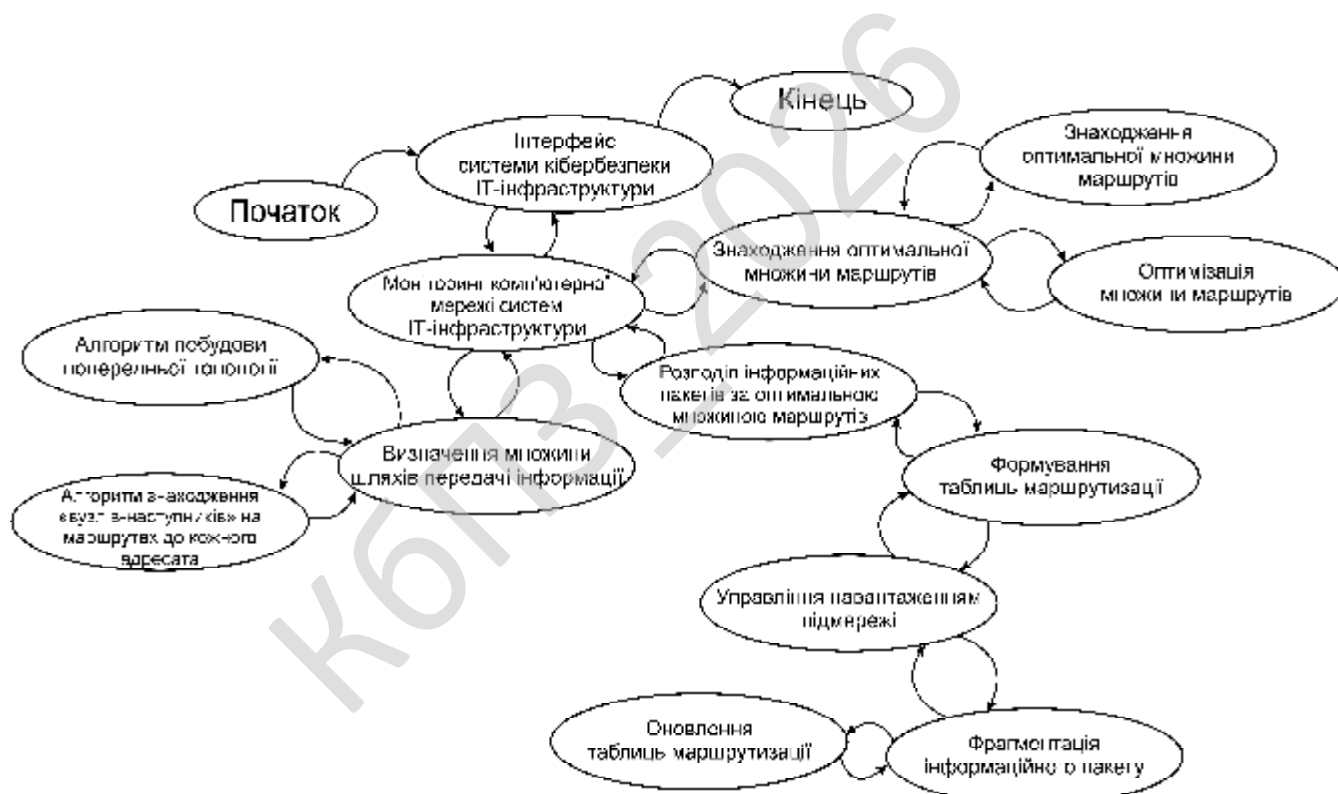


Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

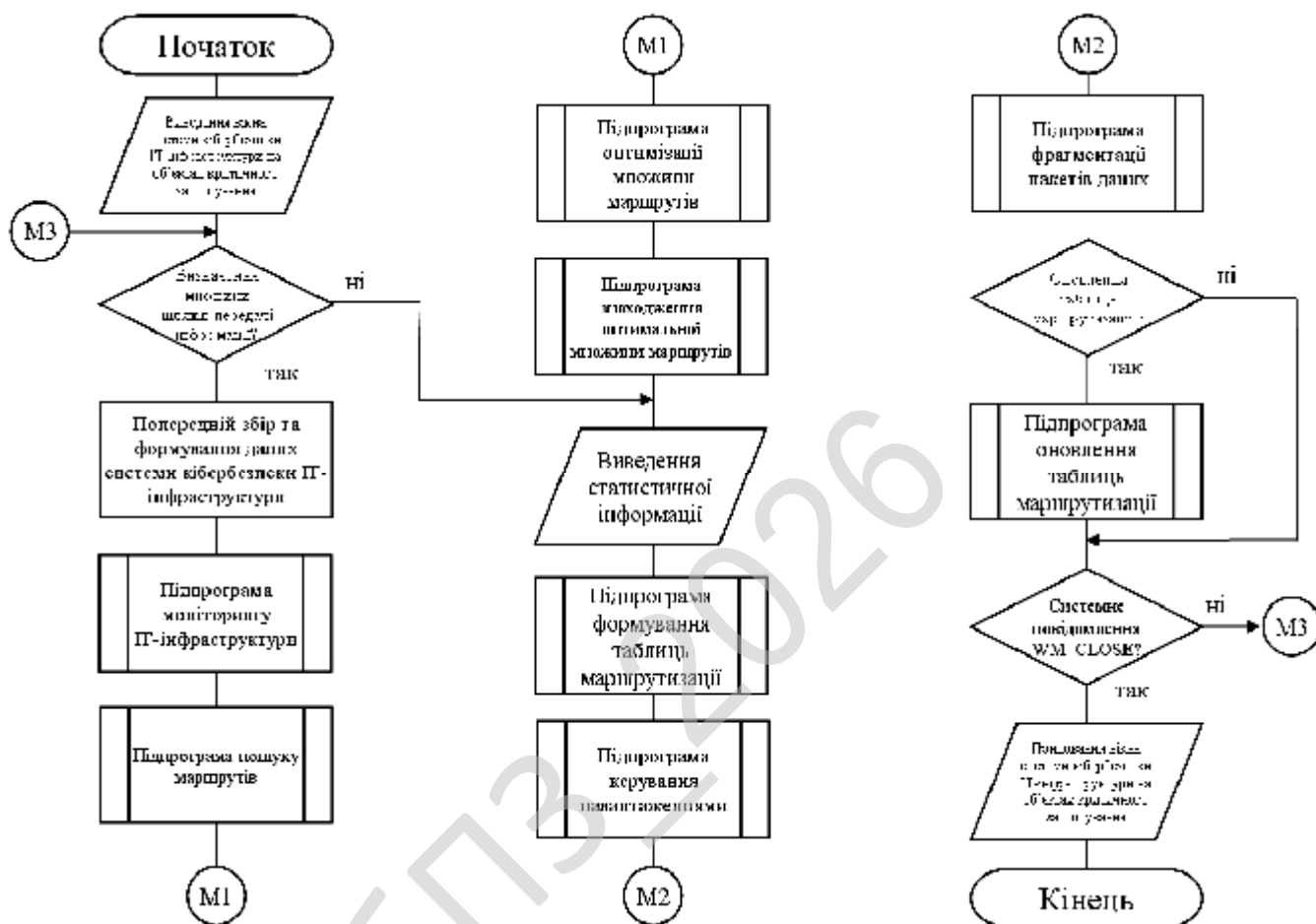


Рисунок 4.1 – Блок-схема основної програми

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням

тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

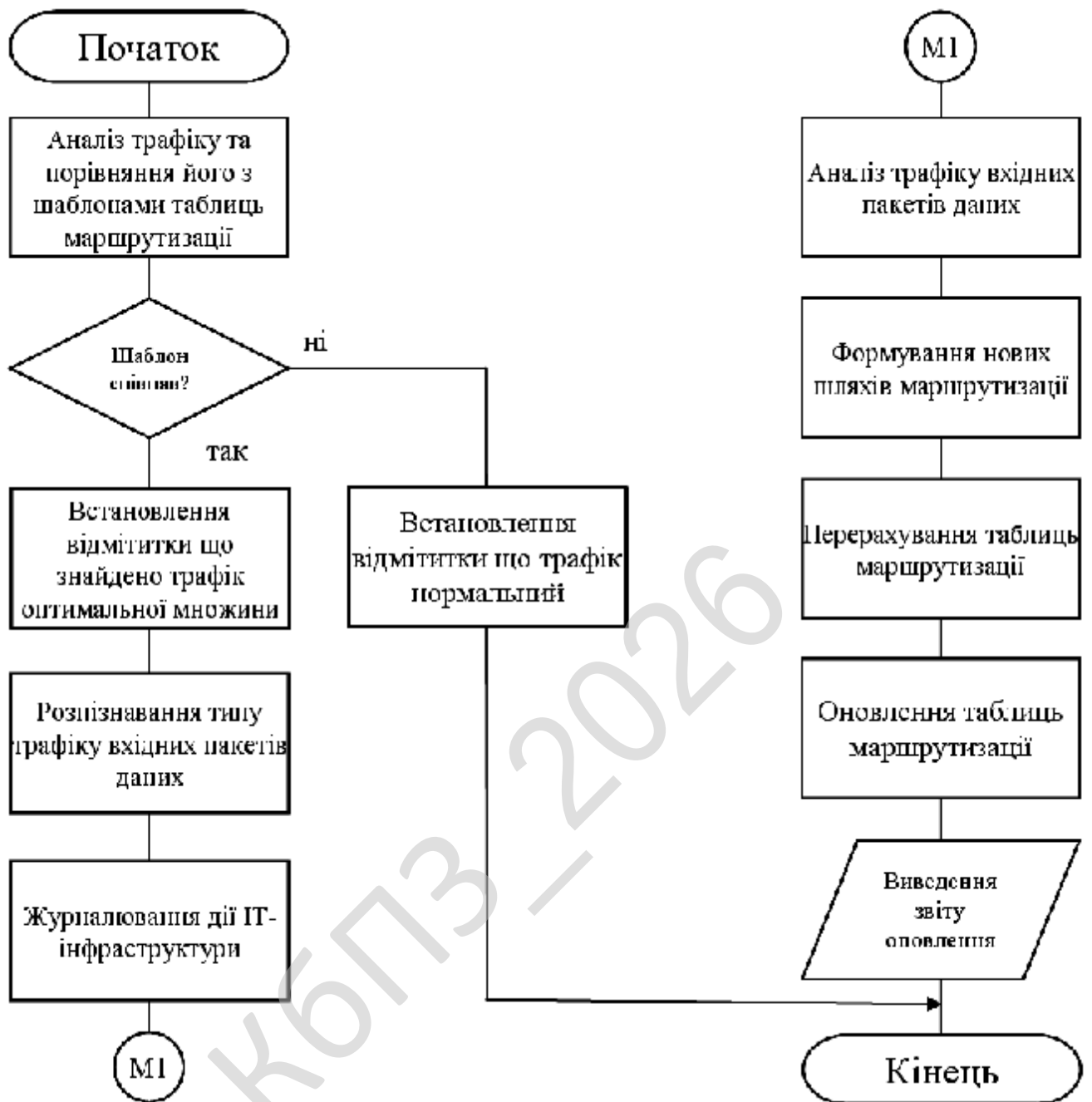


Рисунок 4.2 – Блок-схема роботи підпрограми

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці.

Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший.

Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого».

Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовани ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених

мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є

важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься

до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

– на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;

– на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;

– на третьому етапі вибираються методи і інструменти управління виявленим ризиком;

– на четвертому етапі проводиться безпосереднє управління ризиком;

– на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

PostgreSQL (вимовляється «Пост-грес-К'ю-ель», або «постгрес») об'єктно-реляційна система керування базами даних (СКБД).

Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації.

PostgreSQL – широко розповсюджена система керування базами даних з відкритим сирцевим кодом. Прототип був розроблений в Каліфорнійському університеті Берклі в 1987 році під назвою POSTGRES, після чого активно розвивався і доповнювався. В червні 1990 року з'явилась друга версія із переробленою системою правил маніпулювання та роботи з таблицями, у 1991 році – третя версія, із доданою підтримкою одночасної роботи кількох менеджерів збереження, покращеним механізмом запитів і доповненою системою внутрішніх правил. В цей час POSTGRES використовувався для реалізації великих систем, таких як: система аналізу фінансових даних, пакет моніторингу функціональності потоків, база даних відстеження астероїдів, система медичної інформації, кілька географічних систем. POSTGRES також використовувався як навчальний інструмент в кількох університетах. 1992 року POSTGRES став головною СКБД наукового комп'ютерного проекту Sequoia 2000. 1993 року кількість користувачів подвоїлась. Стало зрозуміло, що для підтримки й подальшого розвитку необхідні великі витрати часу на дослідження баз даних, тому офіційно проект Берклі було зупинено на версії 4.2. 1994 року Andrew Yu і Jolly Chen додали інтерпретатор мови SQL, вдосконалили сирцевий код і виклали в Інтернеті свою реалізацію під назвою Postgres95. 1996 року програмний продукт було перейменовано на PostgreSQL із початковою версією 6.0. Подальшою підтримкою й розробкою займається група спеціалістів у галузі баз даних, які добровільно приєдналися до цього проекту.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Основні можливості, функції

Функції дозволяють виконувати деякий код безпосередньо сервером бази даних. Ці функції можуть бути написані на SQL, який має деякі примітивні програмні оператори, такі як галуження та цикли. Але гнучкішою буде функція написана на одній із мов програмування, з якими PostgreSQL може працювати. До таких мов належать:

Вбудована мова, яка зветься PL/pgSQL, подібна до процедурної мови PL/SQL компанії Oracle.

Мови розробки сценаріїв: PL/Perl, PL/Python, PL/Tcl, PL/Ruby, PL/sh.

Класичні мови програмування C, C++, Java (за допомогою PL/Java).

Функції можуть виконуватись із привілеями користувача, який її викликав, або із привілеями користувача, який її написав.

Основні можливості, індекси

У PostgreSQL є підтримка індексів наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створити нові типи індексів.

Основні можливості, багатоверсійність (MVCC)

PostgreSQL підтримує одночасну модифікацію БД декількома користувачами за допомогою механізму Multiversion Concurrency Control (MVCC). Завдяки цьому виконуються вимоги ACID, і практично відпадає потреба в блокуванні зчитування.

Основні можливості, типи даних

PostgreSQL підтримує великий набір вбудованих типів даних:

1. Числові типи.
2. Цілі.
3. Грошовий тип.
4. Символьні типи довільної довжини.
5. Двійкові типи (включаючи BLOB).
6. Типи «дата/час».
7. Булевий тип.

8. Перерахування.
9. Геометричні примітиви.
10. Мережеві типи.
11. UUID-ідентификатор.
12. XML-дані.
13. JSON-дані.
14. Масиви.
15. OID-типи.

Крім того, користувач може самостійно створювати нові необхідні йому типи та програмувати для них механізми індексування за допомогою GiST.

Основні можливості, об'єкти користувача

PostgreSQL може бути розширено користувачем для власних потреб практично в будь-якому аспекті. Є можливість додавати власні:

1. Перетворення типів.
2. Типи даних.
3. Домени (для користувача типи з самого початку з накладеними обмеженнями).
3. Функції (включаючи агрегатні).
4. Індокси.
5. Оператори (включаючи перевизначення вже існуючих).

Основні можливості, успадкування

Таблиці можуть успадковувати характеристики та набори полів від інших таблиць (батьківських). При цьому дані, які додаються до породженої таблиці, автоматично будуть брати участь (якщо це не вказано окремо) в запитах до батьківської таблиці. Цей функціонал в поточний час не є повністю завершеним. Однак він достатній для практичного використання.

Основні можливості, тригери

Тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, що перевіряє доданий запис

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

на відповідність певним умовам. Тригери можна писати різними мовами програмування. Вони пов'язані з визначеною таблицею. Множинні тригери виконуються в алфавітному порядку.

Інші можливості

1. Дотримання принципів ACID.
2. Відповідність стандартам ANSI SQL-92 і SQL-99.
3. Підтримка запитів з OUTER JOIN, UNION, UNION ALL, EXCEPT і під запитів.
4. Послідовності.
5. Контроль цілісності.
6. Реплікація.
7. Загальні табличні вирази й рекурсивні запити.
8. Аналітичні функції.
9. Підтримка Unicode (UTF-8).
10. Підтримка регулярних виразів у стилі Perl.
11. Вбудована підтримка SSL і Kerberos.
12. Протокол поділюваних блокувань.
13. Завантажувані розширення, підтримують SHA1, MD5, XML і іншу функціональність (API відкритий).
14. Засоби для генерації сумісного з іншими системами SQL-коду та імпорту з інших систем.

Надійність

Згідно з результатами автоматизованого дослідження різного ПЗ на предмет помилок, у вихідному коді PostgreSQL було знайдено 20 проблемних місць на 775 000 рядків вихідного коду (в середньому, одна помилка на 39 000 рядків коду). Для порівняння:

1. MySQL – 97 проблем, одна помилка на 4 000 рядків коду.
2. FreeBSD – 306 проблем, одна помилка на 4 000 рядків коду.
3. Linux (тільки ядро) – 950 проблем, одна помилка на 10 000 рядків коду.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Розглянемо розробку проекту в контрольованому середовищі (PRojects IN Controlled Environments – PRINCE) – це метод управління проектами. Метод включає в себе управління, контроль і організацію проекту. «PRINCE2» – це другий реліз зазначеного методу, що є зареєстрованим торговим знаком, державної торгово–промислової палати (Office of Government Commerce – OGC), незалежного підрозділу державного казначейства Сполученого Королівства.

PRINCE2 походить від більш раннього методу PROMPT та методу проектного управління PRINCE, який був розроблений у 1989 р. Центральним телекомунікаційним та комп'ютерним агентством (Central Computer and Telecommunications Agency – CCTA), як державний стандарт Великобританії з управління проектами у сфері інформаційних технологій (IT). Незабаром зазначений стандарт почали використовувати за межами IT сфери. У 1996 р. PRINCE2 був визнаний універсальним методом управління проектами. PRINCE2 став надзвичайно популярним і зараз де-факто є стандартом проектного управління в Великобританії.

Остання версія була випущена у 2009 р., як результат проекту оновлення Prince2:2009 державної торгово–промислової палати Великобританії.

PRINCE2:2009 Оновлення: З 2006 р. метод переглядався, а 16 Червня, 2009 р. був опублікований під назвою «Оновлення PRINCE2:2009». Назва «PRINCE2» (замість «PRINCE3» чи схожой) використано задля демонстрації збереження головних принципів методу.

Все ж таки, це найбільш суттєвий перегляд методу з 1996 р. з метою його адаптації до мінливого бізнес середовища, спрощення і «полегшення», а також кращої інтеграції з іншими методами державної торгово–промислової палати Великобританії (ITIL, P3O, P3M3, MSP, etc.). Головна різниця між релізом 2009 р. та більш ранніми релізами полягає в появі двох довідників замість одного. «Управління проектами, використовуючи PRINCE2» (Managing Projects Using PRINCE2) – оновлена методика управління, що призначена для менеджерів проектів, для яких управління проектами є щоденною діяльністю. «Стратегічне

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

управління проектами, використовуючи PRINCE2» (Directing Projects Using PRINCE2) – новий документ для Керівника проектів і Ради управління проектами, тобто спонсорів/замовників проектів.

Екзамен як з Теоретичної так і з Практичної частини буде базуватися на новому довіднику «Управління проектами» та не буде включати матеріали з довідника «Стратегічне управління проектами». Прохідний бал для Теоретичного екзамену залишиться не змінним. Прохідний бал для Практичного екзамену збільшиться з 50 % до 55 %. Тривалість екзамену з Практичної частини буде зменшено з 3 годин до 2,5 годин.

PRINCE2 – це структурований підхід до управління проектами, який спрямований на управління проектами в рамках чітко визначеної організаційної структури. PRINCE2 описує процедури координації людей та завдань в проекті, як створювати/планувати проект та що робити, якщо проект потребує внесення змін через невідповідність фактичного стану виконання проекту плану його виконання. Кожний процес зазначено з ключовими вхідними та вихідними даними, а також з специфічними цілями та завданнями, які необхідно виконати, що загалом дає можливість контролю відхилень від плану.

Поділ методу на частини, що підлягають управлінню, забезпечує ефективний контроль ресурсів, завдяки чому можливо здійснювати контрольований та організований моніторинг проекту. PRINCE2 забезпечує єдину термінологію для всіх учасників проекту. Різноманітні ролі управлінців та зони відповідальності повністю описані та можуть бути адаптовані відповідно до складності проекту та компетенцій організації.

Помилки використання

Іноді PRINCE2 вважають не доцільним для дуже маленьких проектів через обсяг роботи, необхідної для створення та оновлення документів, протоколів та реєстрів. Досить часто це трапляється через нерозуміння, які частини PRINCE2 використовувати, адже PRINCE2 можливо повністю масштабувати.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Звичайно все це чудово, але сприйняття PRINCE2 погіршене тим, що для кожного ефективного проекту, який використовує методологію PRINCE2 існує певна кількість не ефективних проектів, які використовують, так звану, методологію PINO (PRINCE In Name Only – дослівно, PRINCE лише в назві).

Загальними проблемами є:

- Стадія початку проекту опускається/проштовхується і організація переходить відразу до створення Документу ініціювання проекту (Project initiation document – PID).

- Рада проекту не є ефективною – ескалації уходять в «чорну діру» тощо.

- Відсутній Опис продукту (Product description – PD), без якого управління якістю в PRINCE2 не працює.

- Припустимі відхилення не встановлюються, або встановлюються лише для часу виконання та вартості – чітко не визначено, в якій мірі повноваження передані виконавцям або коли необхідно використовувати ескалацію (піднімати рівень прийняття рішень).

- Проект фактично складається з однієї стадії (або «фази» використовуються, щоб уникнути Огляду виконання стадії завершення (End Stage Review)).

- Документи ініціювання проекту фактично копії-паст (copy-paste – копіювати – вставити) Документу ініціювання проекту з останнього проекту – їм не слідують і вони створюються тому що «потрібно».

- Низький рівень залучення «бізнесу», адже PRINCE2 вважається «технічним» методом – виходи (продукти) проекту можуть не призвести до бажаних результатів.

- Успіхи та невдачі використання PRINCE2 були однією з причин створення Моделі зрілості PRINCE2 (PRINCE2 Maturity Model – P2MM).

Модель зрілості PRINCE2. Модель зрілості PRINCE2 описує набір Сфер Ключових Процесів (Key Process Areas – KPAs), які необхідні для ефективного впровадження та використання PRINCE2 в організації. Довідник PRINCE2 описує

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

управління проектом і не включає опис процесів впровадження PRINCE2 в організації, в той час як P2MM описує саме процеси впровадження PRINCE2.

P2MM описує ключові практики, які додаються до процесів та складових PRINCE2 з метою забезпечення повторюваного використання методу (Рівень 2 KPAs), а також ключові практики необхідні для впровадження методу (Рівень 3 KPAs), як стандартного бізнес-процесу управління проектами, що включає: призначення власника (3.1), адаптацію методу (3.2), навчання (3.3), інтеграцію з іншими системами управління (3.4), механізм забезпечення якості (3.5).

Огляд методу

Блок-схема процесів PRINCE2. Стрілки означають потоки інформації. PRINCE2 – це процесуальний метод управління проектами на відміну від адаптивних методів, таких як Scrum. PRINCE2 2009 визначає 40 окремих активностей (завдань) і об'єднує їх у сім процесів.

Початок проекту.

На цьому етапі визначаються учасники проектної команди та готується короткий опис проекту (опис цілей проекту та комерційне обґрунтування проекту). Відповідно до загальних засад методу на цьому ж етапі планується наступний етап проекту. Після завершення зазначених робіт Рада проекту має погодити наступний етап проекту – ініціювання проекту.

Головні завдання включають: призначення керівника проекту та менеджера проекту, визначення та призначення команди управління проектом, створення короткого опису проекту, визначення методу управління проектом, планування наступного етапу проекту (ініціювання).

Ініціювання проекту

Цей етап базується на результатах виконання завдань етапу початку проекту. Короткий опис проекту розширюється до бізнес плану (Business case). Точки контролю якості проекту узгоджуються з точками контролю стану виконання проекту. Створюється план виконання наступного етапу проекту.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Результати етапу виносяться на погодження Ради проекту з метою погодження продовження роботи над проектом.

Головні завдання включають: планування показників якості, створення плану проекту, деталізація бізнес плану та ризиків, визначення точок контролю проекту, створення Документу ініціювання проекту (Project Initiation Document).

Стратегічне управління проектом

Цей процес визначає яким чином Рада проекту (що уособлює роль спонсору проекту) буде загалом контролювати проект. Рада проекту дозволяє/санкціонує етап ініціювання проекту, а також сам проект. Стратегічне управління також визначає яким чином Рада проекту має погоджувати план етапів, включаючи погодження будь-якого плану етапу, що може змінити існуючий план проекту у зв'язку з будь-якими змінами строків або ключових показників проекту.

Також визначається, яким чином Рада проекту може змінити план проекту, способи закриття проекту.

Головні завдання включають: засоби погодження ініціювання проекту, погодження проекту, погодження етапу або вилучення етапу проекту, підтвердження завершення проекту.

4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозіумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операцій з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128 раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання за модулем 2^{32} .

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання за модулем 2^{32} . Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму шифрування є його вразливість до атак «на пов'язаних ключах» (англ. Related-key attack). Через простий розклад ключів кожен ключ має 3 еквівалентних ключа. Це означає, що ефективна довжина ключа складає всього 126 біт [3] [4], тому даний алгоритм не слід використовувати в якості геш-функції.

Опис алгоритму

Вихідний текст розбивається на блоки по 64 біта кожен. 128-бітний ключ K ділиться на чотири 32-бітних підключа $K[0]$, $K[1]$, $K[2]$ і $K[3]$. На цьому підготовчий процес закінчується, після чого кожен 64-бітний блок шифрується протягом 32 циклів (64 раундів) за нижченаведеним алгоритмом. [5]

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Припустимо, що на вхід n -го раунду ($1 \leq n \leq 64$) надходять права й ліва частини (L_n, R_n) , тоді на виході n -го раунду будуть ліва й права частини (L_{n+1}, R_{n+1}) , які обчислюються за такими правилами:

$$L_{n+1} = R_n.$$

Якщо $n = 2 * i - 1$ для $1 \leq i \leq 32$ (непарні раунди), то:

$$R_{n+1} = L_n(\{ [R_n 4] K [0] \} \{ R_n i * \delta \} \{ [R_n 5] K [1] \}).$$

Якщо $n = 2 * i$ для $1 \leq i \leq 32$ (парні раунди), то:

$$R_{n+1} = L_n(\{ [R_n 4] K [2] \} \{ R_n i * \delta \} \{ [R_n 5] K [3] \}).$$

Де

$X \oplus Y$ – операція додавання чисел X і Y за модулем 2^{32} .

$X \oplus Y$ – побітове виключне АБО» (XOR) чисел X і Y , яке в мові програмування Cі позначається як $X \wedge Y$

$X \ll Y$ і $X \gg Y$ – операції побітового зсуву числа X на Y біт вліво й вправо відповідно.

Константа δ була виведена з Золотого перерізу:

$$\delta = (-1) * 2^{31} = 2654435769 = 9E3779B9_{16}.$$

У кожному раунді константа множиться на номер циклу i . Це було зроблено для запобігання простих атак, заснованих на симетрії раундів.

Також очевидно, що в алгоритмі шифрування TEA немає як такого алгоритму розкладу ключів. Замість цього в непарних раундах використовуються підключі $K [0]$ та $[1]$, у парних – $K [2]$ і $[3]$.

Так як це блочний шифроалгоритм, де довжина блоку 64-біт, а довжина даних може бути не кратна 64-біт, значення всіх байтів, які доповнюють блок до кратності в 64-біт, встановлюється в 0x01.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Розділу обрання групи; Розділу виведення результату роботи системи.

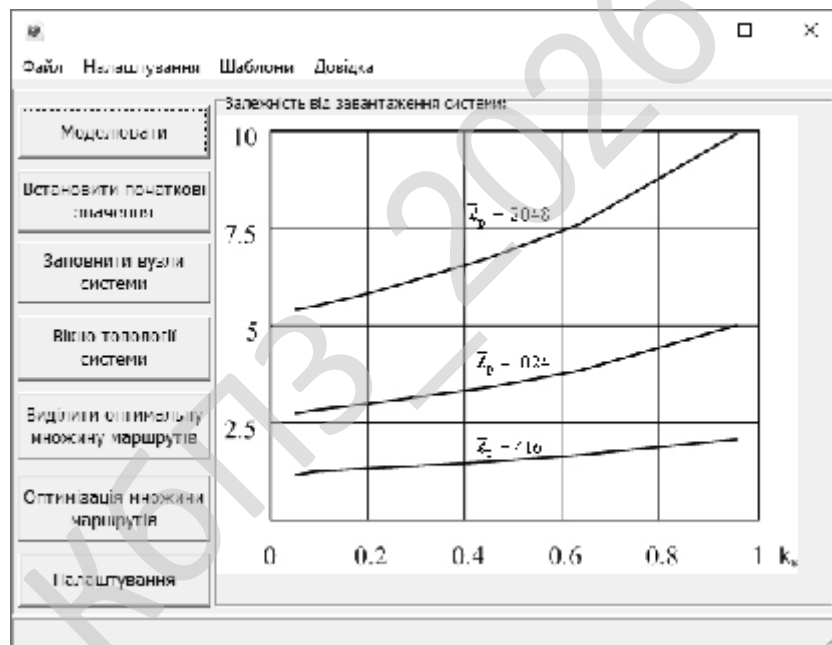


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких

помилки, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

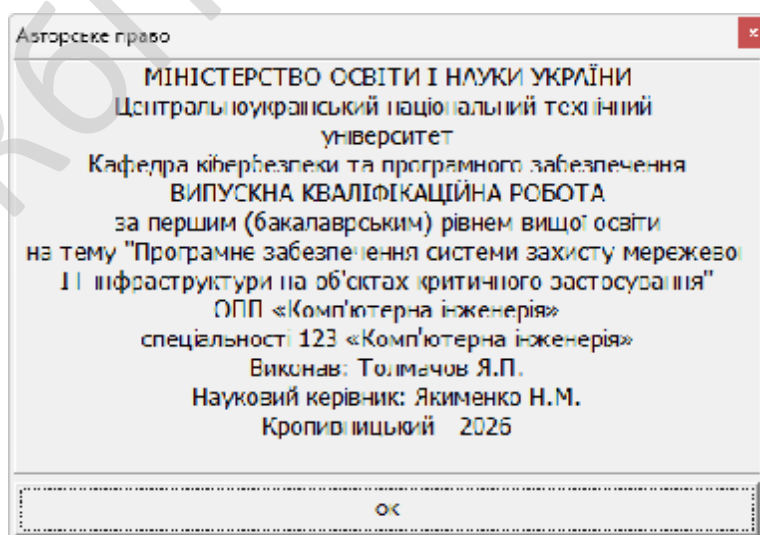


Рисунок 5.2 – Авторське право

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс.
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій.
- Помилки інтерфейсу.
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.

- Помилки характеристик (необхідна ємність пам'яті і т.д.).

- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

(тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ).

Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи захисту мережевої IT-інфраструктури на об'єктах критичного застосування.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем захисту мережевої IT-інфраструктури на об'єктах критичного застосування.
- Досліджена система захисту мережевої IT-інфраструктури на об'єктах критичного застосування.
- На основі отриманих результатів досліджень створена програмна реалізація системи захисту мережевої IT-інфраструктури на об'єктах критичного застосування.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання захисту мережевої IT-інфраструктури на об'єктах критичного застосування.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

системи захисту мережевої ІТ-інфраструктури на об'єктах критичного застосування. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ТЕА.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 p.
2. Massimo Bertaccini. Cryptography Algorithms. Packt Publishing. 2022. 358 p.
3. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.
4. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.
5. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
6. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
7. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
8. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
9. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
10. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
11. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
12. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.
13. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.— К.: Видавничий центр “Кафедра”, 2025.— 320 с.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

14. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

15. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

16. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

17. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

18. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

19. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

20. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security

Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

21. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

22. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 170–188.

23. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

24. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

25. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

26. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

27. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

28. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

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

30. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

31. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: *Rajakumar, G., Du, K.L., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

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

33. Смірнов О.А., Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІІШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

34. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп'ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

35. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

36. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп’ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв’язку, 2023, вип. 2(72), С. 170-178.

37. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

38. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

39. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

40. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв’язку*, 2022, № 3(69). С. 93-98.

41. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

42. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

43. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, Cracow, Poland, 22-25 September 2021. P. 414-418

44. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

45. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

46. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings Volume 2805*, 2020, Pages 44-58.

47. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

48. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

49. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

50. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

51. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

52. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

53. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

54. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

55. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable*

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

56. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

57. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

58. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

59. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660.

60. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

61. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

62. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

63. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

64. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», 2019 *IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

65. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

К6ПЗ-2026

					ВКРБ-123.26.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94