

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 кібербезпеки для захисту
даних за рахунок використання Endpoint Security”

Виконав здобувач вищої освіти
IV курсу, групи КБ-22-2
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Кравченко Д.О.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнова Т.В.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 125 "Кібербезпека"
Освітньо-професійна (освітньо-наукова) програма "Кібербезпека"

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

Кравченку Денису Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security*

2. Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 129-02 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 кібербезпеки для захисту даних за рахунок використання Endpoint Security*

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 кібербезпеки в промислову експлуатацію.

6. Висновки

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

Структурна схема системи кібербезпеки 1 аркуш

Функціональна схема системи кібербезпеки 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

Смірнова Т.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Кравченко Д.О.
(прізвище та ініціали)

АНОТАЦІЯ

Кравченко Д.О. Програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

Метою розробки є програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

Результат роботи – програмна реалізація системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

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: кібербезпека, Endpoint Security

ABSTRACT

Kravchenko D.O. Software for a cybersecurity system for data protection using Endpoint Security. 125 Cybersecurity. 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 that is intended for a cybersecurity system for data protection using Endpoint Security.

The purpose of the development is software for a cybersecurity system for data protection using Endpoint Security.

The result of the work is a software implementation of a cybersecurity system for data protection using Endpoint Security.

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 a PC with OS Windows 10/11.

The program was developed in the Python environment.

Keywords: cybersecurity, Endpoint Security

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.....	5
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 кібербезпеки та мови програмування.....	19
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	41
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	41
4.2 Захист розробленого програмного забезпечення.....	67
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	70
6 ОСНОВНІ ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78

					ВКРБ-125.26.0034.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	Кравченко Д.О.					Б	1	84
<i>Перев.</i>	Смірнова Т.В.							
<i>Н.контр.</i>	Коваленко А.С.					ЦНТУ КБ-22-2		
<i>Затв.</i>	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

КМ	—	комп'ютерна мережа
КСАЗ	—	комплексна система антивірусного захисту
МЕ	—	міжмережний екран
ПЗ	—	програмне забезпечення
ПК	—	персональний комп'ютер
ACL	—	Access Control List
FTP	—	File Transfer Protocol
http	—	HyperText Transfer Protocol
IoA	—	індикатори атак
IoC	—	індикатори компрометації
POP3	—	Post Office Protocol Version 3
SMTP	—	Simple Mail Transfer Protocol
VLAN	—	Virtual Local Area Network

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Захист кінцевих точок (Endpoint Security) був основним компонентом програм кібербезпеки, але його роль швидко розширюється. У 2026 році кінцеві точки більше не обмежуються корпоративними ноутбуками та настільними комп'ютерами. Тепер вони включають персональні пристрої, системи підрядників, хмарні робочі навантаження та віддалені кінцеві точки, що працюють далеко за межами традиційних мережевих меж.

Оскільки організації впроваджують розподілені моделі роботи та хмарні архітектури, кінцеві точки залишаються однією з найпоширеніших точок входу для кіберінцидентів. Водночас регуляторні органи та аудитори дедалі частіше очікують, що організації продемонструють, як ризики кінцевих точок виявляються, контролюються та відстежуються з часом. Це робить захист кінцевих точок не лише технічною проблемою, а й пріоритетом управління та дотримання вимог.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

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

- Огляд існуючих систем для захисту даних за рахунок використання Endpoint Security.
- Дослідження системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.
- Програмна реалізація системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для захисту даних за рахунок використання Endpoint Security.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ_2026

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Визначення кінцевої точки значно розширилося за останні роки. У більшості організацій сьогодні кінцеві точки включають:

- Ноутбуки та настільні комп'ютери співробітників.
- Мобільні пристрої, що використовуються для роботи.
- Ресурси «Принеси свій власний пристрій» (BYOD).
- Пристрої підрядника та сторонніх виробників.
- Віртуальні машини та хмарні робочі навантаження.

Це розширення створює проблему з видимістю. Багато організацій мають труднощі з веденням точної інвентаризації кінцевих точок, розумінням того, кому належить кожен пристрій, та визначенням того, чи послідовно застосовуються засоби контролю безпеки.

Без чіткої видимості, забезпечення дотримання політик захисту кінцевих точок стає реактивним, а не превентивним. Організації, які ставляться до управління кінцевими точками як до статичної активності, зіткнуться зі зростаючим впливом кіберризиків та ризиків, пов'язаних з дотриманням вимог.

1.2 Область застосування

Ринок засобів безпеки кінцевих точок (Endpoint Security) стрімко зростає останніми роками. Він зросте з 18,58 млрд доларів у 2025 році до 20,79 млрд доларів у 2026 році зі сукупним річним темпом зростання (CAGR) 11,9%. Зростання за історичний період можна пояснити зростанням кількості кінцевих пристроїв у підприємствах, збільшенням частоти кібератак, розширенням

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

моделей віддаленої та гібридної роботи, зростанням впровадження мобільності підприємств, посиленням регуляторної уваги до захисту даних.

Очікується, що ринок безпеки кінцевих точок швидко зростатиме протягом наступних кількох років. У 2030 році він зросте до 32,52 мільярда доларів США зі складним річним темпом зростання (CAGR) 11,8%. Зростання в прогнозований період можна пояснити збільшенням інвестицій у передову аналітику загроз, зростанням попиту на уніфіковані платформи управління кінцевими точками, розширенням безпеки для віддалених працівників, зростаючим впровадженням автоматизованого реагування на інциденти, посиленням інтеграції безпеки кінцевих точок із ширшими екосистемами безпеки. Основні тенденції в прогнозований період включають зростання впровадження захисту кінцевих точок на основі штучного інтелекту, зростання використання рішень для виявлення та реагування на кінцеві точки, зростаючий попит на хмарну безпеку кінцевих точок, розширення платформ безпеки мобільних пристроїв, посилену увагу до доступу до кінцевих точок з нульовою довірою.

Очікується, що зростання кількості кінцевих точок стимулюватиме зростання ринку безпеки кінцевих точок у найближчі роки. Кінцева точка – це віддалений обчислювальний пристрій, який регулярно взаємодіє з підключеною мережею. Прикладами кінцевих точок є смартфони, настільні комп'ютери та ноутбуки. Безпека кінцевих точок – це стратегія, розроблена для запобігання використанню кіберзлочинцями кінцевих точок або точок входу пристроїв кінцевих користувачів, таких як ПК, ноутбуки та мобільні пристрої, для зловмисних цілей.

Наприклад, у вересні 2024 року німецька неурядова організація IoT Analytics GmbH повідомила, що до кінця 2023 року було підключено 16,6 мільярда пристроїв Інтернету речей, що на 15% більше, ніж у 2022 році. IoT Analytics також прогнозує подальше зростання на 13%, досягнувши 18,8 мільярда пристроїв до кінця 2024 року. Крім того, очікується, що кількість підключених

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

пристроїв Інтернету речей досягне 40 мільярдів до 2030 року, що є значним збільшенням з моменту останнього оновлення ринку у 2023 році. Таким чином, зростання кількості кінцевих точок стимулює зростання ринку безпеки кінцевих точок.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-125.26.0034.00.00.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

Великі компанії, що працюють на ринку безпеки кінцевих точок, зосереджуються на інноваційних продуктах, таких як trend vision one, щоб збільшити доходи на своєму ринку. Мета платформи trend vision one – надати підприємствам інструменти, необхідні для розробки надійних кіберпрограм та проактивного управління ризиками, запобігаючи будь-яким інцидентам та порушенням. Наприклад, у липні 2023 року японська компанія з кібербезпеки trend Micro Incorporated запустила trend vision one. Платформа спрощує ІТ-операції та операції з безпеки, поєднуючи захист, виявлення та реагування. Рішення наступного покоління інтегрує центри обробки даних, сервери, хмарні робочі навантаження, кінцеві точки користувачів, а також запобігання, виявлення та реагування. Мета цього комплексного рішення – допомогти клієнтам у процесі трансформації їхніх ІТ-систем.

У серпні 2023 року Cisco Systems Inc., американська технологічна компанія, що спеціалізується на розробці, виробництві та продажу мережевого обладнання, програмного забезпечення та послуг, придбала Oort Inc. за нерозголошену суму. Це придбання має на меті покращити можливості Cisco у сфері кібербезпеки шляхом посилення захисту ідентифікаційних даних, покращення пропозицій Security Cloud та надання передових інструментів для виявлення загроз на основі ідентифікаційних даних та реагування на них. Oort Inc. – це американська компанія-розробник програмного забезпечення, яка пропонує платформу безпеки на базі штучного інтелекту, розроблену для допомоги організаціям у захисті від загроз ідентифікаційним даним.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Основними компаніями, що працюють на ринку безпеки кінцевих точок, є VMware Inc.; McAfee Corp.; Sophos Group plc; Palo Alto Networks Inc.; Panda Security SL; FireEye Inc.; Microsoft Corporation; CrowdStrike Holdings Inc.; Trend Micro Incorporated; Kaspersky Lab AO; Cisco Systems Inc.; SentinelOne Inc.; AVG Technologies CZ sro; BlackBerry Limited; Check Point Software Technologies Ltd.; Broadcom Inc.; Comodo Security Solutions Inc.; F-Secure Corporation; Bitdefender SRL; Webroot Inc.; Malwarebytes Corporation; Cylance Inc.; Fortinet Inc.; SonicWall Inc.; Barracuda Networks Inc.; WatchGuard Technologies Inc.; Proofpoint Inc.; Zscaler Inc.; Forcepoint LLC; Ivanti Inc.

Північна Америка буде найбільшим регіоном на ринку безпеки кінцевих точок у 2025 році. Очікується, що Азіатсько-Тихоокеанський регіон буде найшвидше зростаючим регіоном у прогнозований період. Регіони, що охоплюються звітом про ринок безпеки кінцевих точок: Азіатсько-Тихоокеанський регіон, Південно-Східна Азія, Західна Європа, Східна Європа, Північна Америка, Південна Америка, Близький Схід, Африка. Країни, що охоплюються звітом про ринок безпеки кінцевих точок: Австралія, Бразилія, Китай, Франція, Німеччина, Індія, Індонезія, Японія, Тайвань, Росія, Південна Корея, Велика Британія, США, Канада, Італія, Іспанія.

Тарифи впливають на ринок безпеки кінцевих точок, підвищуючи витрати, пов'язані з імпортом обладнання для кібербезпеки, безпечним мережевим обладнанням та передовою аналітичною інфраструктурою, що використовується разом із платформами захисту кінцевих точок. Великі підприємства в Північній Америці та Європі найбільше постраждали через залежність від інтегрованих глобальних стеків безпеки, тоді як Азіатсько-Тихоокеанський регіон стикається з тиском на витрати при масштабуванні розгортання безпеки кінцевих точок. Ці тарифи збільшують загальні витрати на кібербезпеку та уповільнюють цикли оновлення. Водночас вони заохочують ширше впровадження хмарних рішень для захисту кінцевих точок, локалізованого надання послуг та програмно-орієнтованих архітектур безпеки.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Звіт про дослідження ринку безпеки кінцевих точок є одним із серії нових звітів, що надає статистику ринку безпеки кінцевих точок, включаючи розмір світового ринку галузі безпеки кінцевих точок, регіональні частки, конкурентів з часткою ринку безпеки кінцевих точок, детальні сегменти ринку безпеки кінцевих точок, тенденції та можливості ринку, а також будь-які додаткові дані, які можуть знадобитися вам для успіху в галузі безпеки кінцевих точок. Цей звіт про дослідження ринку безпеки кінцевих точок надає повне уявлення про все, що вам потрібно, з поглибленим аналізом поточного та майбутнього сценарію галузі.

Безпека кінцевих точок стосується процесу забезпечення безпеки або захисту таких пристроїв, як ноутбуки, мобільні телефони, настільні комп'ютери та планшети, від кібератак та шкідливих загроз. За допомогою цього підприємства можуть захистити свої пристрої, які використовуються співробітниками для робочих цілей. Безпека кінцевих точок використовується для захисту організацій від дедалі небезпечніших кіберзагроз.

Основними компонентами безпеки кінцевих точок є програмне забезпечення та послуги. Програмне забезпечення - це набір інструкцій, програм або даних, які використовуються для роботи комп'ютерів та виконання певних завдань. Основні розгортання здійснюються локально та в хмарі. Основні розміри підприємств - це великі підприємства та малі та середні підприємства. Різні вертикалі включають банківську справу, фінансові послуги та страхування (BFSI), телекомунікації та IT, державне управління та оборону, охорону здоров'я та науки про життя, виробництво, роздрібну торгівлю та електронну комерцію, а також інші вертикалі.

Ринок безпеки кінцевих точок складається з доходів, отриманих суб'єктами господарювання, забезпечуючи гібридну хмарну безпеку та безпеку SaaS. Вартість на цьому ринку є «вартістю на вході заводу», тобто вартістю товарів, проданих виробниками або творцями товарів, чи то іншим суб'єктам господарювання (включаючи виробників, оптовиків, дистриб'юторів та роздрібних торговців), чи безпосередньо кінцевим споживачам. Вартість товарів

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

на цьому ринку включає супутні послуги, що продаються творцями товарів.

Вибір рішення для захисту кінцевих точок може бути складним. В ідеалі вам не слід шукати одне антивірусне програмне забезпечення, яке може захистити ваші кінцеві точки. Натомість вам потрібна система безпеки корпоративного рівня, яка може забезпечити безпеку кінцевих точок, одночасно підвищуючи загальний рівень безпеки.

Іншими словами, він повинен добре інтегруватися з вашою існуючою інфраструктурою безпеки та діяти як доповнення, якщо не як невід'ємна частина.

Ось деякі з найкращих рішень для безпеки кінцевих точок, які ви можете розглянути у 2026 році. Ми ретельно відібрали їх на основі кількох факторів, включаючи функції та рейтинги .

SentinelOne Singularity Endpoint Security

SentinelOne Singularity Endpoint Security – це уніфікована платформа, розроблена для забезпечення комплексного контролю вашого стану кібербезпеки на всіх кінцевих точках. Вона пропонує захист від усіх кіберзагроз у масштабах підприємства, забезпечуючи надійне запобігання, виявлення та реагування на них на різних пристроях.

Централізований підхід гарантує, що ваші команди безпеки можуть підтримувати нагляд і контроль, одночасно оптимізуючи робочі процеси та прискорюючи реагування на нові загрози, такі як шкідливе програмне забезпечення та програми-вимагачі.

Особливості:

- Динамічне виявлення пристроїв для ідентифікації некерованих кінцевих точок у режимі реального часу.
- Краща прозорість та підзвітність зі зменшенням кількості хибнопозитивних результатів у різних операційних системах.
- Можливості швидкого реагування для виправлення та відкату кінцевих точок одним клацанням миші.
- Найкраще у своєму класі виявлення та реагування на кінцеві точки

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

(EDR) для точного вирішення загроз.

- Автоматична кореляція подій для реконструкції хронології загроз.
- Централізоване віддалене керування через консоль RemoteOps для швидкого реагування на загрози.

- Основні проблеми безпеки кінцевих точок, які усуває SentinelOne
SentinelOne Singularity™ Endpoint Security вирішує кілька критичних проблем кібербезпеки, з якими може зіткнутися ваш бізнес:

- Кінцеві точки часто є основними цілями для зловмисників. SentinelOne допомагає захистити ці пристрої, зменшуючи їхню вразливість до порушень.

- Багато організацій намагаються забезпечити видимість усієї поверхні атаки, що вирішується завдяки комплексному аналізу всіх кінцевих точок, включаючи некеровані пристрої, та покращеному обізнанню про стан безпеки.

- Традиційні методи реагування можуть бути трудомісткими та неефективними. Цю проблему вирішують можливості швидкого реагування.

- Хибнопозитивні результати, що впливають на ефективність виявлення, зменшуються завдяки автономним, комбінованим рішенням для безпеки кінцевих точок EPP+EDR.

- Проблема різнорідних систем вирішується за допомогою єдиної централізованої платформи, яка об'єднує дані та робочі процеси.

Symantec Endpoint Security Complete

Symantec Endpoint Security Complete забезпечує безпеку кінцевих точок для підприємств. Він захищає мобільні пристрої протягом усього ланцюжка атак. Symantec може адаптуватися до різних загроз, а також пропонує функції управління безпекою на основі штучного інтелекту.

Особливості:

- Захист від шкідливого програмного забезпечення та вторгнень: сканує та блокує підозрілі файли або дії, прагнучи запобігти несанкціонованим змінам.

- Запобігання втраті даних (DLP): моніторинг та контроль конфіденційних даних для зменшення випадкового або навмисного розкриття.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- Аналіз загроз на основі штучного інтелекту: використовує методи машинного навчання для виявлення незвичайних закономірностей та швидшого визначення загроз.
- Захист хмари та мережі: генерує аналітичні дані з хмарних джерел та аналітичні дані на рівні мережі для підвищення безпеки кінцевих точок.
- Централізоване керування: пропонує єдину консоль для налаштування політик, відстеження інцидентів та створення звітів безпеки.
- Звітність про відповідність: Надає інструменти для виконання нормативних вимог; також генерує звіти про відповідність.
- Покриття мобільних пристроїв: Розширює захист мобільних кінцевих точок та забезпечує стабільну безпеку.

Sophos Intercept X – Sophos

Sophos Intercept X – це платформа безпеки кінцевих точок, розроблена для захисту даних на різних пристроях, у хмарних середовищах та на серверах. Налаштування та керування системою здійснюється через хмарну платформу безпеки Sophos, яка також виконує функції керованого виявлення та реагування (MDR).

Особливості:

- Виявлення та реагування на кінцеві точки: моніторинг загроз на різних пристроях та в різних середовищах, пропонуючи інструменти для виявлення та аналізу підозрілої активності.
- Захист від шкідливого програмного забезпечення, експлойтів та програм-вимагачів: захищає від різних типів атак, допомагаючи зменшити ризик компрометації системи.
- Хмарне керування: для налаштування та адміністрування покладається на хмарну консоль, яка також може слугувати службою керованого виявлення та реагування (MDR).
- Керування мобільними кінцевими точками: включає опції для контролю мобільних пристроїв, що допомагає забезпечити узгодженість політик та захисту.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

FortiClient – Fortinet

FortiClient від Fortinet – це «Fabric Agent», по суті програмне забезпечення для кінцевих точок, що працює на кінцевих пристроях. Його використовують організації з віддаленими працівниками, яким потрібні безпечні мережеві з'єднання на основі принципів нульової довіри. FortiClient – одне з найкращих рішень для безпеки кінцевих точок.

Особливості:

- «Fabric Agent» для даних кінцевих точок: встановлюється на кінцеві пристрої для збору інформації про їхній стан та рівень безпеки.
- Віддалений доступ із нульовою довірою: використовує кроки перевірки для захисту з'єднань віддалених працівників, щоб запобігти несанкціонованому доступу.
- Автоматизовані відповіді: Вживає заходів щодо скомпрометованих кінцевих точок, щоб стримувати проблеми до їх поширення.
- Веб-безпека та фільтрація контенту: допомагає контролювати доступ до сайтів та контенту, зменшуючи вплив шкідливих матеріалів.
- Управління програмним забезпеченням та інвентарем: відстежує встановлені програми та пристрої, допомагаючи в обслуговуванні та оновленнях.

Trend Vision One

Trend Vision One – це рішення для захисту кінцевих точок, яке пропонує кілька рівнів захисту для всіх можливих етапів атаки на кінцеві пристрої. Його цілодобова MDR-система гарантує виявлення будь-якої загрози та вжиття заходів. Trend Vision One охоплює широкий спектр операційних систем та платформ Linux, які використовують кілька серверних систем.

Особливості:

- Консолідований захист кінцевих точок, серверів та робочих навантажень.
- Високоточне машинне навчання.
- Поведінковий аналіз.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

- Блокування систем командування та управління (C&C).
- Запобігання втраті даних (DLP).
- Відкат програм-вимагачів.

Harmony

Harmony – це рішення для захисту кінцевих точок від Check Point Software. Воно допомагає організаціям захистити свої дані в електронній пошті, веб-сайтах, додатках та корпоративному доступі. Harmony може зменшити складність інфраструктури, зменшити ризики хмарних мереж, а також надає гібридні інструменти безпеки.

Особливості:

- Захист від загроз штучного інтелекту: працює на всіх кінцевих точках, блокуючи шкідливі програми та експлойти в режимі реального часу.
- Уніфікована консоль: для перегляду та керування політиками безпеки кінцевих точок з однієї панелі.
- Консолідовані інструменти безпеки: вони розроблені для зниження загальних витрат на захист за допомогою інтегрованої платформи.
- Управління загрозами: Виявлення та визначення пріоритетів вразливостей, перш ніж їх можна буде використати.
- Забезпечує безпечний віддалений доступ: для користувачів, які знаходяться поза межами офісу, забезпечує безпечне підключення до корпоративних ресурсів.

Microsoft Defender

Microsoft Defender для кінцевих точок – це ще одне рішення для захисту кінцевих точок. Воно може запобігати, виявляти, досліджувати та реагувати на загрози в корпоративних мережах. Воно може інтегруватися з іншими службами Microsoft та забезпечувати цілісний захист кінцевих точок.

Особливості:

- Моніторинг поведінкових кінцевих точок: він виявляє аномальну активність за допомогою вбудованих поведінкових датчиків.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Хмарна аналітика: вона використовується для виявлення складних загроз за допомогою масштабованої хмарної аналітики Microsoft.
- Аналітика загроз: інформує вас про найновіші загрози та збирає інформацію про них.
- Розслідування та сповіщення: вони виконуються автоматично, що зменшує потребу в ручному реагуванні на загрози.
- Інтеграція з Microsoft 365 та службами Azure: об'єднує засоби захисту на пристроях Microsoft та Azure, хмарних службах і програмах.

Trellix

На відміну від більшості рішень для безпеки кінцевих точок, які пропонуються як єдиний продукт, Trellix пропонує комплекс найкращих рішень для безпеки кінцевих точок. Цей комплекс продуктів пропонує все: від багаторівневих підходів до захисту мобільних пристроїв, контролю програм та змін тощо.

Особливості:

- Багаторівневий захист від загроз: він використовується для блокування шкідливого програмного забезпечення, експлойтів та інших атак на кінцевих точках.
- Судово-медичне розслідування: призначене для глибокого аналізу інцидентів безпеки та їх першопричин.
- Мобільна безпека: пропонує функції для захисту смартфонів і планшетів як частину пакету кінцевих точок.
- Оркестрація політик: забезпечує послідовне дотримання правил безпеки в усій організації.
- Обмін інформацією про загрози: Поставляється з інтегрованою платформою обміну, щоб бути в курсі останніх загроз.
- Контроль програм та змін: блокує несанкціоновані програми та системні зміни на кінцевих точках.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

CrowdStrike Falcon

CrowdStrike Falcon забезпечує захист кінцевих точок від різних атак, включаючи шкідливе програмне забезпечення та загрози нульового дня. Він використовує платформу захисту кінцевих точок на основі штучного інтелекту (EPP), яка спирається на сканування пам'яті для запобігання поширенню загроз.

Особливості:

- Захист від загроз штучного інтелекту: він зупиняє шкідливе програмне забезпечення та атаки нульового дня на кінцеві точки.
- Блокування та стримування загроз: мінімізує потребу в ручному втручанні системи безпеки шляхом автоматичного блокування та карантину загроз.
- Відстеження загроз: відстежує загрози на кінцевих точках для моніторингу активності та поведінки зловмисника.
- Видимість: Додає видимість щодо вразливостей кінцевих точок, щоб зменшити поверхню атаки.
- Проактивне полювання на загрози: воно виявляє приховані або сплячі загрози.
- Захист ідентифікаційних даних: виявляє та зупиняє атаки на облікові записи користувачів на основі облікових даних.
- Панель керування інцидентами: спрощує розслідування та реагування на інциденти безпеки.

Cortex XDR

Cortex XDR впроваджує захист усього підприємства, аналізуючи дані з усіх можливих джерел для запобігання атакам. Він може усунути сліпі зони безпеки, спростити операції безпеки для скорочення середнього часу реагування (MTTR) і, зрештою, зменшити витрати шляхом консолідації операцій безпеки.

Особливості:

- Виявлення загроз машинного навчання: використовує дані з кінцевих точок, мереж та хмарних джерел для раннього виявлення загроз.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Управління інцидентами: спрощує реагування та розслідування в єдиному інтерфейсі.
- Аналіз першопричин: він може швидко визначити, як сталася та поширилася атака.
- Судово-медичні дані: це може забезпечити збір судово-медичних даних для аналізу атак та поведінки шкідливого програмного забезпечення.
- Полювання на загрози: Він може виявляти загрози на кінцевих точках, у мережах та хмарних середовищах, щоб виявити приховані та невідомі фактори.

Як вибрати правильне рішення для безпеки кінцевих точок?

Вибір правильного рішення для захисту кінцевих точок має вирішальне значення для захисту вашої організації від кіберзагроз. З огляду на різноманітність доступних варіантів, важливо оцінити кілька ключових факторів, щоб переконатися, що рішення відповідає вашим конкретним потребам та вимогам безпеки.

Ось структурований підхід, який допоможе вам прийняти обґрунтоване рішення:

1. Комплексний захист. Шукайте рішення для забезпечення безпеки кінцевих точок, які пропонують багаторівневі функції безпеки, включаючи антивірус, захист від шкідливого програмного забезпечення, брандмауери та засоби виявлення загроз. Забезпечте надійний захист від усіх можливих типів загроз.

2. Можливості виявлення та реагування. Оцініть здатність рішення виявляти загрози в режимі реального часу та автоматично реагувати. Ефективні можливості виявлення та реагування на кінцеві точки (EDR) та розширеного виявлення та реагування (XDR) можуть значно скоротити час пом'якшення наслідків інцидентів.

3. Легкість управління. Зважте на інтерфейс керування та загальну зручність використання рішення. Централізована панель інструментів, яка спрощує завдання моніторингу та управління, є важливою для ефективності.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

4. Масштабованість. Переконайтеся, що рішення може масштабуватися разом із зростанням вашої організації. Воно повинно бути здатним обробляти зростаючу кількість кінцевих точок без шкоди для продуктивності чи безпеки. Найкраще рішення для безпеки кінцевих точок – це те, яке працює для вашої організації та не відстає від неї.

5. Сумісність. Перевірте сумісність з існуючими системами та програмним забезпеченням. Рішення для безпеки кінцевих точок повинно безперешкодно інтегруватися з вашим поточним ІТ-середовищем, щоб мінімізувати перебої в роботі.

6. Відповідність. Переконайтеся, що рішення для безпеки кінцевих точок відповідає галузевим вимогам. Це особливо важливо для організацій у регульованих секторах, таких як фінанси та охорона здоров'я.

Висновок

З огляду на приголомшливе зростання кіберзагроз, захист вашого підприємства від загроз є важливішим, ніж будь-коли. Правильні рішення для безпеки кінцевих точок, такі як SentinelOne Singularity™ Endpoint Security, не лише захищають ваші цінні дані, але й дають вашій команді змогу впевнено працювати, знаючи, що вони захищені від потенційних порушень.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Python – це об'єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

– Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.

– Об'єктно-орієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення веб-сайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веб-розробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

- Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

- Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

- Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

- Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

- Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

– Неефективна для пам'яті. Ця мова програмування використовує великий обсяг пам'яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам'яті.

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам'ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок,

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веб-розробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

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 кібербезпеки 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 кібербезпеки 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-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Безпека кінцевих точок стосується низки методів і технологій, розроблених для захисту пристроїв, що підключаються до вашої мережі, від кіберзагроз. Ці пристрої називаються кінцевими точками та включають ноутбуки, настільні комп'ютери, мобільні телефони, планшети, сервери та віртуальні середовища, які співробітники використовують щодня.

Безпека кінцевих точок – це не окрема система чи програмне забезпечення. Вона складається з кількох елементів, які допомагають впроваджувати та підтримувати безпеку кінцевих точок. До них належать, але не обмежуються:

1. Брандмауери.
2. Шифрування даних.
3. Системи виявлення та запобігання вторгненням (IDPS).
4. Виявлення та реагування на кінцеві точки (EDR).
5. Керування виправленнями.
6. Контроль доступу.

Потреба в безпеці кінцевих точок

Рішення для захисту кінцевих точок стають рятівним колом сучасного бізнесу. Користувачі можуть вийти за межі корпоративної інфраструктури безпеки, а сучасні загрози надзвичайно складні та різноманітні. Традиційних рішень безпеки, таких як антивірус та брандмауери, недостатньо для забезпечення захисту периметра. Вам потрібні найкращі інструменти захисту кінцевих точок, щоб захиститися від мінливих та динамічних кібератак. Організації також зростають та масштабуються, а це означає, що вартість захисту їхніх мереж зростає.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Локальна безпека кінцевих точок потребує тривалого налаштування та не є найкращим вибором для користувачів з обмеженим бюджетом. Крім того, вона обмежується підтримкою безпеки лише власних серверів. Щойно ви переходите до хмари, ваша інфраструктура стає вразливою. Пропозиції щодо безпеки кінцевих точок SentinelOne охоплюють хмарні приміщення та послуги. Вони забезпечують швидке реагування на загрози, виявлення інцидентів та цілодобовий захист кінцевих точок. Хороші пакети захисту кінцевих точок пропонують швидке встановлення виправлень, надійний контроль доступу та безпечні віддалені пристрої.

Багато пристроїв Інтернету речей не мають необхідних заходів шифрування. Встановивши EDR для керування, моніторингу та зменшення вразливостей кінцевих точок, ви можете залишатися в курсі подій та регулювати авторизований доступ до пристроїв. Якщо вам цікаво, «Що таке рішення для безпеки кінцевих точок», ви будете раді дізнатися, що це спеціальні інструменти, які можуть сегментувати ваші мережі та ізолювати проблеми в них. Електронна пошта є одним із найпоширеніших шлюзів для проникнення кіберзлочинців. Шкідливі файли надсилаються між пристроями та мережами, що становить загрозу для безпеки кінцевих точок. Потужні інструменти безпеки кінцевих точок можуть помістити загрози в карантин та забезпечити додаткові функції безпеки, такі як фільтрація контенту, архівування електронної пошти, захист від фішингу, блокування шкідливого програмного забезпечення тощо. Документування здатності вашої організації обробляти конфіденційні дані та дотримуватися вимог також може допомогти уникнути значних штрафів від регуляторних органів та інших юридичних витрат. Безпека кінцевих точок необхідна для забезпечення постійної відповідності стандартам, що постійно змінюються, таким як SOC 2, HIPAA, NIST Benchmark та інші нормативні рамки.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Ключові ризики, які організації повинні враховувати під час захисту кінцевих точок

Кінцеві точки залишаються основним вектором атаки для широкого спектру кіберінцидентів. Хоча методи атак продовжують розвиватися, кілька ризиків, пов'язаних з кінцевими точками, постійно з'являються в розслідуваннях порушень та висновках регуляторних органів:

- Шкідливе програмне забезпечення та програми-вимагачі, що надходять через вкладення електронної пошти або скомпрометовані завантаження.
- Крадіжка облікових даних, що виникає із заражених або погано захищених пристроїв.
- Фішингові атаки, що використовують доступ до кінцевих точок для нестандартного просування.
- Непатчені операційні системи та програми, що розкривають відомі вразливості.
- Надмірні дозволи та доступ локального адміністратора на кінцевих точках.

Ці ризики посилюються в розподілених середовищах, де пристрої працюють поза межами корпоративних мереж, а традиційні засоби захисту периметра пропонують обмежений захист. Ефективне вирішення проблеми захисту кінцевих точок вимагає структурованого підходу, який поєднує технічний контроль, управління доступом та постійний нагляд.

Найкращі практики захисту основних кінцевих точок

Ось список найкращих практик захисту кінцевих точок, яких повинні дотримуватися організації:

1. Ведіть повний та точний інвентаризацію кінцевих точок

Ефективний захист кінцевих точок починається зі знання того, які пристрої існують. Організації повинні підтримувати актуальний інвентар, який відображає:

- Тип пристрою та операційна система.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

- Власність та призначений користувач.
- Підключення до корпоративних систем.
- Стан контролю безпеки.

Без цієї базової лінії стає важко оцінити ризики, забезпечити дотримання політик або продемонструвати охоплення засобами контролю під час аудитів кібербезпеки .

2. Забезпечте послідовне впровадження засобів контролю безпеки кінцевих точок

Засоби контролю безпеки слід застосовувати послідовно на всіх керованих кінцевих точках. Це включає:

- Виявлення та реагування кінцевих точок (EDR) або антивірусний захист.
- Шифрування диска для даних у стані спокою.
- Безпечні базові конфігурації.
- Брандмауери на базі хоста.

Узгодженість має значення. Розриви між відділами, місцями розташування або типами пристроїв часто створюють найслабші місця, які використовують зловмисники.

3. Посилити контроль ідентифікації та доступу на рівні кінцевих точок

Захист кінцевих точок тісно пов'язаний із безпекою ідентифікаційних даних. Навіть добре захищений пристрій може стати ризикованим, якщо облікові дані скомпрометовані або доступ надмірно обмежений.

Найкращі практики включають:

- Забезпечення багатфакторної автентифікації (MFA).
- Застосування принципів доступу з найменшими привілеями.
- Обмеження або скасування прав локального адміністратора.
- Регулярно перевіряти доступ.

Організації можуть обмежити вплив компрометації кінцевих точок, зменшивши непотрібний доступ.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

4. Безперервно виправляйте та оновлюйте кінцеві точки

Непатчені кінцеві точки залишаються одним із найпоширеніших джерел зловмисних дій. Організаціям слід встановити чіткі процеси для:

- Оновлення операційної системи.
- Виправлення застосунків.
- Оновлення прошивки та драйверів.

Чіткі процеси оновлення операційної системи, встановлення виправлень для програм та оновлення прошивки є надзвичайно важливими. Автоматизація відіграє тут вирішальну роль, особливо у віддалених та гібридних середовищах, де ручні оновлення ненадійні.

5. Моніторинг активності та поведінки кінцевих точок

Захист кінцевих точок повинен виходити за рамки запобігання та виявляти. Діяльність з моніторингу кібербезпеки допомагає організаціям виявляти:

- Аномальна поведінка.
- Підозрілі схеми доступу.
- Ознаки компромісу.

Раннє виявлення скорочує час простою та обмежує вплив інцидентів до їх ескалації.

6. Захист кінцевих точок у віддалених та гібридних середовищах

Моделі віддаленої та гібридної роботи створюють додаткові труднощі. Пристрої можуть підключатися з незахищених мереж, працювати поза межами корпоративних інструментів моніторингу або використовуватися спільно з іншими користувачами.

Щоб вирішити цю проблему, організаціям слід:

- Застосовуйте політики кінцевих точок незалежно від їх розташування.
- Вимагати безпечної автентифікації для віддаленого доступу.
- Застосовувати умовний доступ на основі положення пристрою.
- Регулярно оцінюйте некеровані або частково керовані пристрої.

Відстеження цієї інформації часто призводить до прогалин під час аудитів та збільшує ризик дотримання вимог.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Поширені недоліки в захисті кінцевих точок, з якими стикаються організації

Незважаючи на значні інвестиції в засоби безпеки, багато організацій стикаються з постійними проблемами захисту кінцевих точок:

- Неповна видимість усіх кінцевих точок.
- Непослідовне застосування контролю.
- Надмірна залежність від ручних процесів.
- Складність у наданні доказів, готових для аудиту.
- Обмежена узгодженість між командами безпеки та відповідності.

Ці прогалини часто виникають через те, що захист кінцевих точок розглядається як суто технічна функція, а не як структурований процес управління кіберризиками.

Посиліть захист кінцевих точок за допомогою підходу, що базується на ризиках

У 2026 році стратегії захисту кінцевих точок повинні вийти за рамки ізольованих інструментів та сповіщень. Підхід, що ґрунтується на оцінці ризиків, зосереджений на:

- Визначення та визначення пріоритетів ризиків, пов'язаних з кінцевими точками.
- Зіставлення контролю з цими ризиками.
- Моніторинг ефективності контролю з плином часу.
- Документування рішень та результатів.

Такий підхід дозволяє організаціям керувати захистом кінцевих точок як частиною ширшої системи управління, а не як сукупністю непов'язаних між собою видів діяльності.

Керування захистом кінцевих точок у великих масштабах

Захист кінцевих точок – це вже не просто захист пристроїв, а управління ризиками в розподіленому, динамічному середовищі. Оскільки кінцевих точок стає більше, а вимоги до відповідності зростають, організаціям потрібні структуровані, повторювані стратегії кібербезпеки, які забезпечують прозорість, узгодженість та підзвітність.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

3.2 Розробка структурної схеми

Багаторівневий захист

Система кібербезпеки для захисту даних за рахунок використання Endpoint Security – сполучає у собі технології для захисту всіх пристроїв на будь-якій платформі від численних, постійно мінливих кіберзагроз. Інструменти контролю, автоматичний захист від експлойтів, система запобігання вторгнень і інші функції істотно знижують імовірність зараження шкідливим ПЗ. Інші технології й процеси виявляють і розпізнають шкідливе ПЗ на робочих місцях, блокують його активність і роблять автоматичний відкіт дій, виконаних шкідливою програмою до її блокування.

Сполучення передових технологій і глибокої аналітики

Унікальний підхід, створений у даній роботі, – це сполучення аналізу «великих даних», машинного навчання й великого експертного досвіду. Все це допомагає успішно протидіяти складним новітнім атакам і забезпечує високий рівень виявлення погроз.

Інтеграція із хмарною базою даних

Мільйони користувачів в усьому світі добровільно й анонімно надають зі своїх комп'ютерів у відповідні бази дані про погрози. Ця хмарна мережа збирає й зберігає величезні обсяги метаданих про підозрілі файли. Такі відомості, що надходять у реальному часі, дозволяють захищати корпоративну мережу від новітніх погроз, у тому числі атак нульового дня.

Виявлення підозрілого поведіння

У системі кібербезпеки для захисту даних за рахунок використання Endpoint Security – і на стадії, що передує виконанню файлу, і під час виконання файлу використовується поведінковий аналіз. Ця функція підтримується хмарною мережею, у якій зберігається величезний обсяг метаданих про підозрілі файли. До запуску файлу виявлення підозрілого поведіння з використанням емуляції відіграє критичну роль у визначенні невідомих і комплексних погроз.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Коли додаток запускається в корпоративній мережі, компонент Аналіз поведінки відслідковує його активність. Помітивши підозріле поведінку, компонент автоматично блокує додаток, а за допомогою компонента Відкриття шкідливих дій може вироблятися автоматичне скасування всіх дій, уже зроблених шкідливим ПЗ. Це ефективний механізм боротьби із програмами-шифрувальниками.

Автоматичний захист від експлойтів

Не існує застосунків або операційних систем, у яких би повністю були відсутні уразливості. Уразливості можуть експлуатуватися шкідливим ПЗ для проникнення в корпоративну мережу, зараження робочих станцій і серверів і порушення бізнес-процесів компанії.

Блокування мережних атак

Кількість погроз, що піддають ризику зараження корпоративні мережі, у тому числі сканування портів, атаки типу «відмова в обслуговуванні» і «переповнення буфера», продовжує рости. Технологія захисту від мережних атак у складі системи кібербезпеки для захисту даних за рахунок використання Endpoint Security – відслідковує підозрілі дії в корпоративній мережі й дозволяє заздалегідь установити сценарій реагування на підозріле поведінку.

Захист загальних папок від шифрування

Система кібербезпеки для захисту даних за рахунок використання Endpoint Security містить механізм протидії, що блокує шифрування загальних файлових ресурсів шкідливим процесом, виконуваним на іншому комп'ютері в тій же мережі. Додаток постійно спостерігає за захищеними загальними папками, відслідковуючи стан файлів, що зберігаються в них. При виявленні шифрування система блокує доступ до сервера для комп'ютера – джерела атаки, зупиняючи процес шифрування й запобігаючи втраті корпоративних даних.

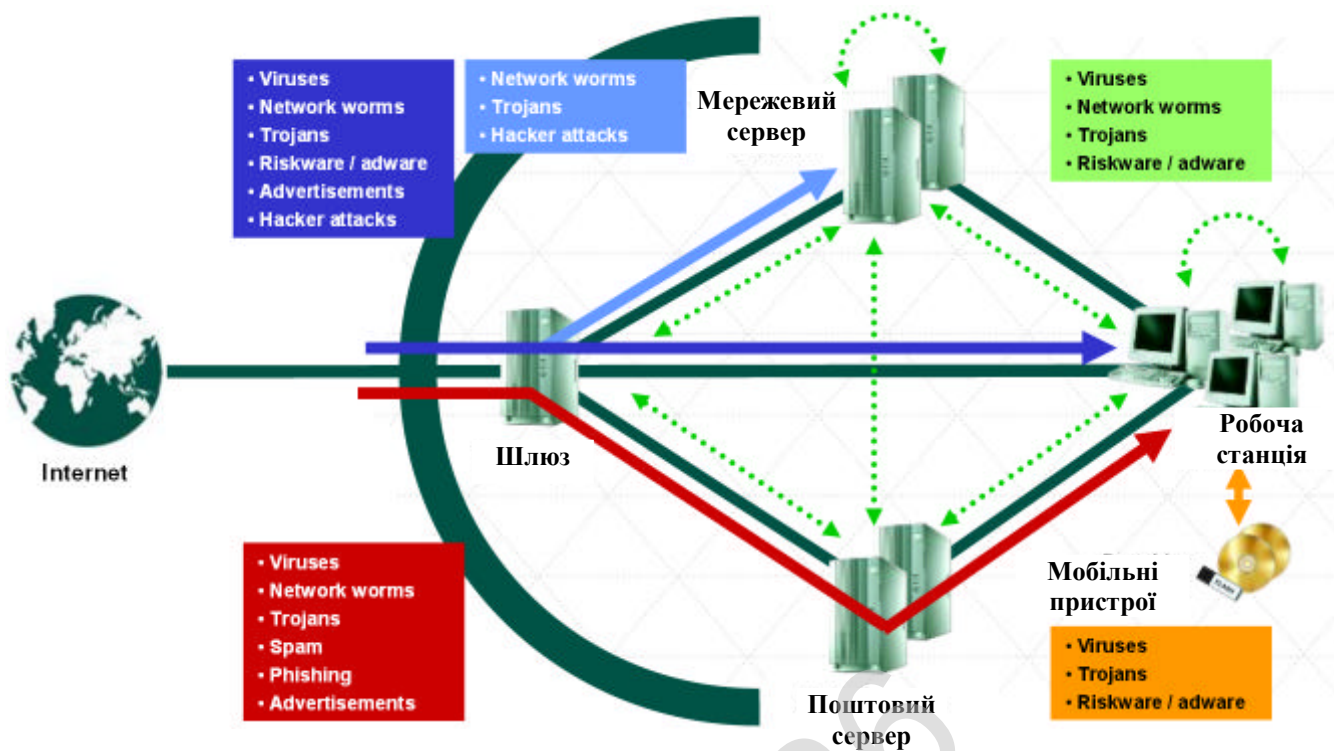


Рисунок 3.1 – Структурна схема системи

Динамічні білі списки

З кожним днем росте кількість програм, потрібних для вашого бізнесу. Відслідковувати, які з потенційно небезпечні, а які немає – непросте завдання. Динамічні білі списки дозволяють системним адміністраторам активувати політикові «Заборона за замовчуванням», що блокує всі програми, за винятком тих, що перебувають у білому списку. Клієнти мають доступ до цієї бази даних і можуть користуватися нею в тому виді, у якому вона є, або адаптувати її до своїх конкретним бізнес-вимогам.

Контроль програм

Коли додаток запускається на сервері або робочому місці співробітника, наш модуль «Моніторинг системи» відслідковує його активність. Помітивши підозріле поведіння, модуль автоматично блокує шкідливі файли, а на робочих станціях «Моніторинг системи» робить автоматичний відкіт дій, виконаних шкідливою програмою до її блокування.

Контроль пристроїв

Контроль пристроїв дозволяє запобігти доступу неавторизованих пристроїв у корпоративну мережу. Система кібербезпеки для захисту даних за рахунок використання Endpoint Security також реєструє всі операції видалення й копіювання, виконувані на знімних USB-пристроях, і управляє правами користувача для операцій читання й запису файлів на CD і DVD-дисках.

Гнучкий контроль підключення до Wi-Fi мереж

Використання недовірених публічних мереж Wi-Fi піддає пристрої й корпоративну мережу ризику. Створення списку довірених мереж спеціально для співробітників дозволяє надавати їм доступ до довірених Wi-Fi мережам і забороняти використання інших мереж.

Крім того, завдяки функції анти-бріджингу Адміністратори можуть забороняти користувачам одночасне встановлення двох мережних з'єднань для запобігання несанкціонованого створення мостів до внутрішньої мережі в обхід засобів захисту периметра. У системі кібербезпеки для захисту даних за рахунок використання Endpoint Security адміністратори також можуть забороняти встановлення одночасних з'єднань із використанням провідної мережі й Wi-Fi.

Захист мобільних пристроїв

Мобільні пристрої використовуються сьогодні в кожній компанії і є потенційними провідниками шкідливого ПЗ у вашу корпоративну мережу. Система кібербезпеки для захисту даних за рахунок використання Endpoint Security – захищає мобільні пристрої від новітніх мобільних погроз. Технологія антифішингу захищає співробітників від підроблених сайтів, а фільтри антиспаму блокують небажані дзвінки й SMS-повідомлення.

Підтримка розповсюджених платформ MDM

Для керування мобільними пристроями (Mobile Device Management, MDM) вам необхідно захисне рішення, інтегроване з усіма основними MDM-платформами. Система кібербезпеки для захисту даних за рахунок використання Endpoint Security підтримує Microsoft Exchange ActiveSync, iOS MDM і Samsung

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

KNOX і дозволяє легко створювати політики для кожної платформи, наприклад обов'язкове шифрування, використання пароля, використання камери, налаштування APN/VPN. Android for Work дає можливість створювати бізнес-профіль, управляти бізнес-додатками й пристроями. Крім того, у рішенні реалізована інтеграція із платформою керування корпоративними мобільними пристроями – VMware AirWatch.

Керування всіма функціями з єдиної консолі

Кожна хвилина, витрачена на адміністрування й складання звітів, – це час, яке можна було б використовувати для рішення більше важливих бізнес-завдань. У рішення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security – входить Security Center (SC) – централізована консоль керування, що дозволяє одержати повне подання про використовувані у вашій корпоративній мережі технології для захисту робочих місць, і управляти ними. KSC спрощує керування мобільними пристроями, ноутбуками, настільними комп'ютерами, файловими серверами й віртуальними машинами, а також генерує звіти.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Висока продуктивність і багаторівневий захист

Захист не повинен сповільнювати роботу ваших систем, тому система кібербезпеки для захисту даних за рахунок використання Endpoint Security впливає на їхню продуктивність. А якщо ви піддалися атаці, рішення дозволяє автоматично скасувати зроблені шкідливі дії, щоб користувачі могли працювати у звичайному режимі.

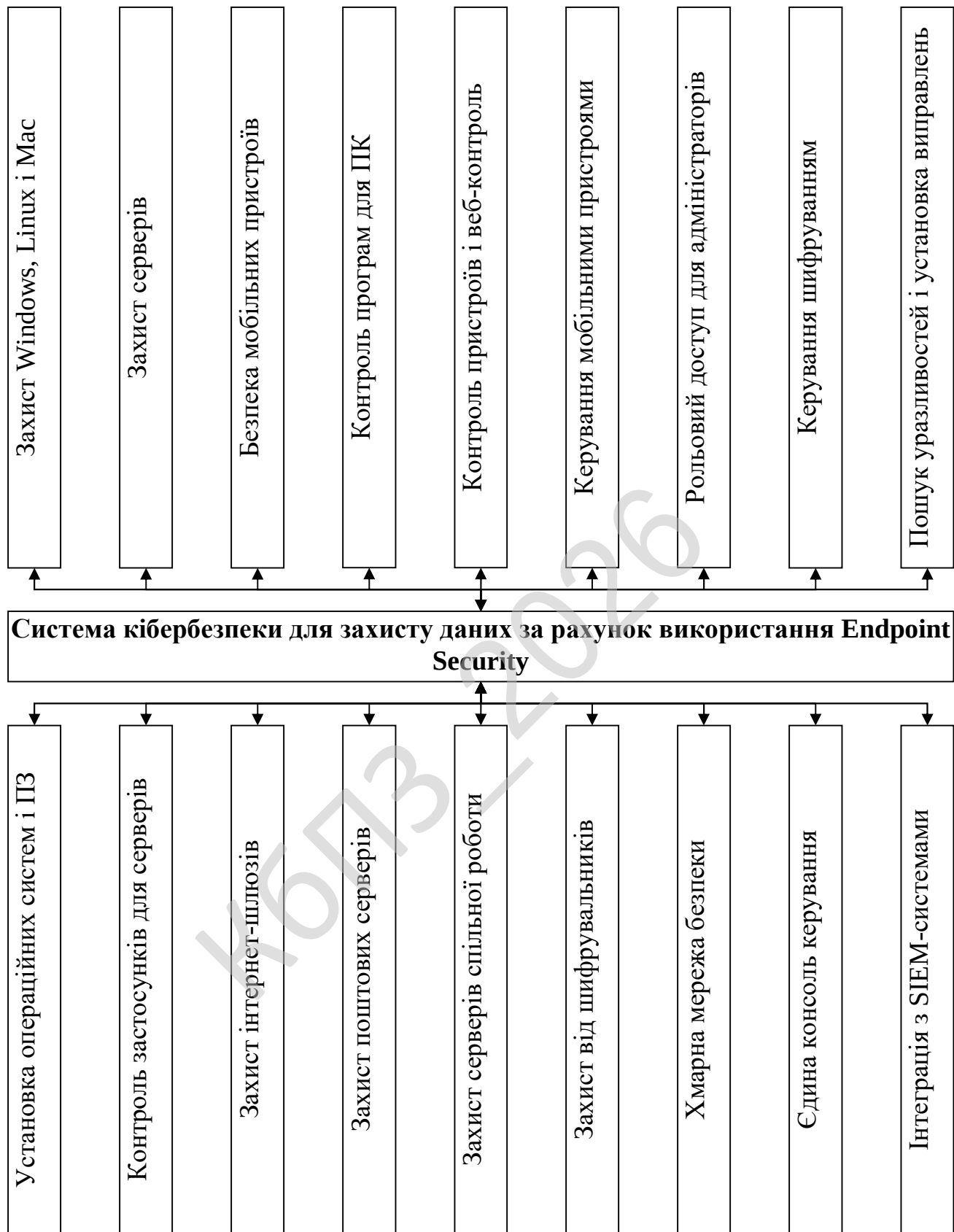


Рисунок 3.2 – Функціональна схема системи

Захисні технології нового покоління

Технології системи кібербезпеки для захисту даних за рахунок використання Endpoint Security, включаючи статичне й динамічне машинне навчання, забезпечують безпека світового класу. Вони скорочують ризик успішних атак на ваш бізнес і захищають всі вузли вашої мережі.

Зручне керування системою безпеки

Рішення дозволяє швидко й просто розгорнути систему захисту, використовуючи широкий набір передвстановлених сценаріїв. Єдина універсальна консоль керування дозволяє ефективно застосовувати задані політики безпеки на кожному вузлі вашої мережі.

Захист вузлів мережі зараз і в майбутньому

Система кібербезпеки для захисту даних за рахунок використання Endpoint Security постійно вивчає погрози й впроваджує новітні технології запобігання атак у свої продукти. Завдяки цьому ваша компанія надійно захищена не тільки від відомих погроз, але й атак, які можуть відбутися в майбутньому.

Значна економія ваших засобів

Застосування різних технологій, у тому числі технологій захисту нового покоління, у складі єдиного продукту дозволяє позбавити вас від схованих витрат. Для захисту всіх вузлів вашої мережі потрібно тільки один продукт із єдиною ліцензією.

Переваги інтеграції

Всі ключові технології, використовувані в рішенні, розроблені усередині системи кібербезпеки для захисту даних за рахунок використання Endpoint Security на власній базі. Додатки й функції в складі рішення тісно інтегровані між собою, що дозволяє забезпечити максимально ефективний і надійний захист ваших систем.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Функціональні можливості системи кібербезпеки для захисту даних за рахунок використання Endpoint Security

Система кібербезпеки для захисту даних за рахунок використання Endpoint Security реалізує наступні функції:

- Захист Windows, Linux і Mac.
- Захист серверів.
- Безпека мобільних пристроїв.
- Контроль програм для ПК.
- Контроль пристроїв і веб-контроль.
- Керування мобільними пристроями.
- Рольовий доступ для адміністраторів.
- Керування шифруванням.
- Пошук уразливостей і установка виправлень.
- Установка операційних систем і ПЗ.
- Контроль застосунків для серверів.
- Захист інтернет-шлюзів.
- Захист поштових серверів.
- Захист серверів спільної роботи.
- Захист від шифрувальників.
- Хмарна мережа безпеки.
- Єдина консоль керування.
- Інтеграція з SIEM-системами.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

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.

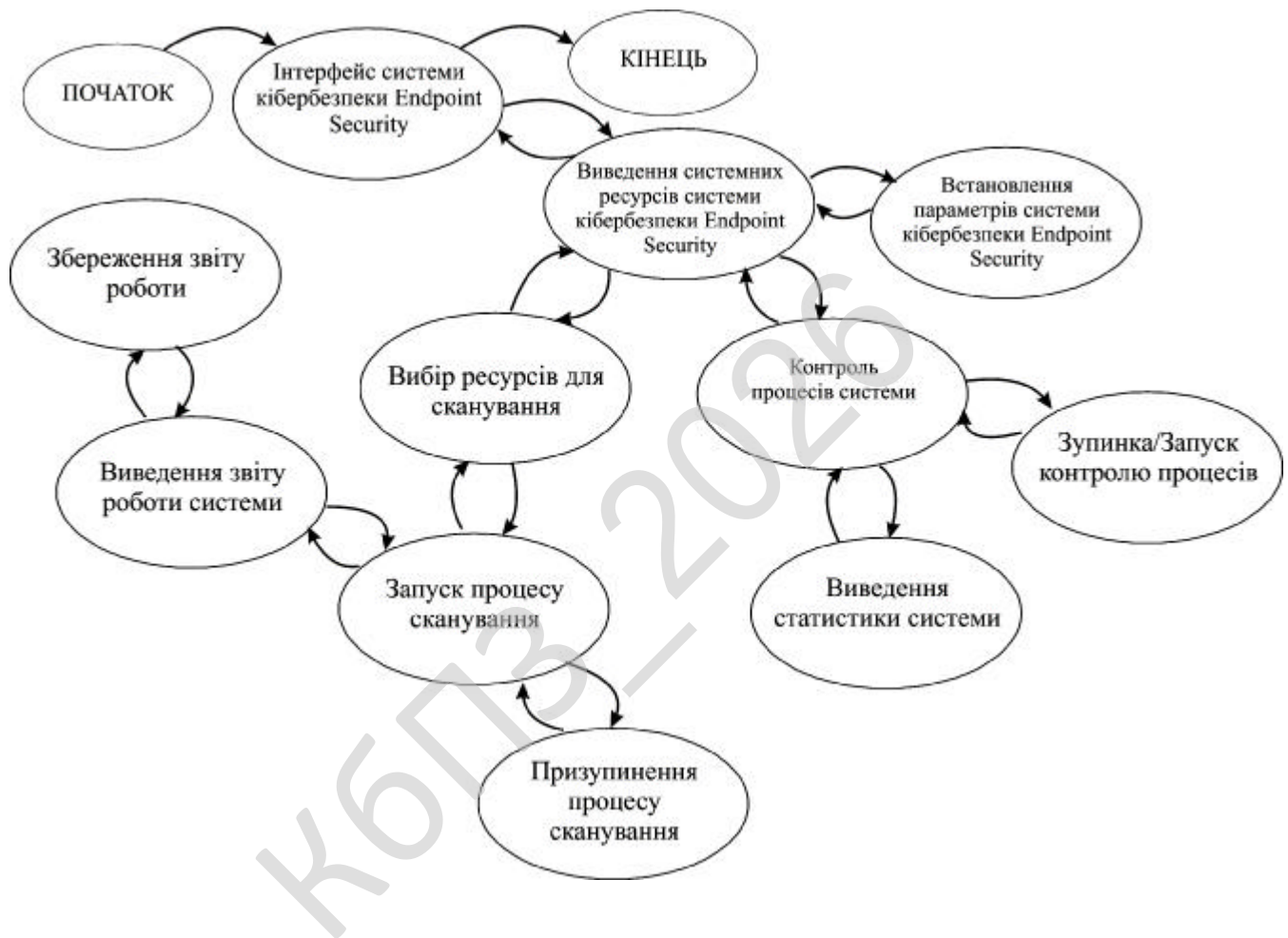


Рисунок 3.3 – Діаграма взаємодії процесів

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.

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).

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.

КБПЗ-2026

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

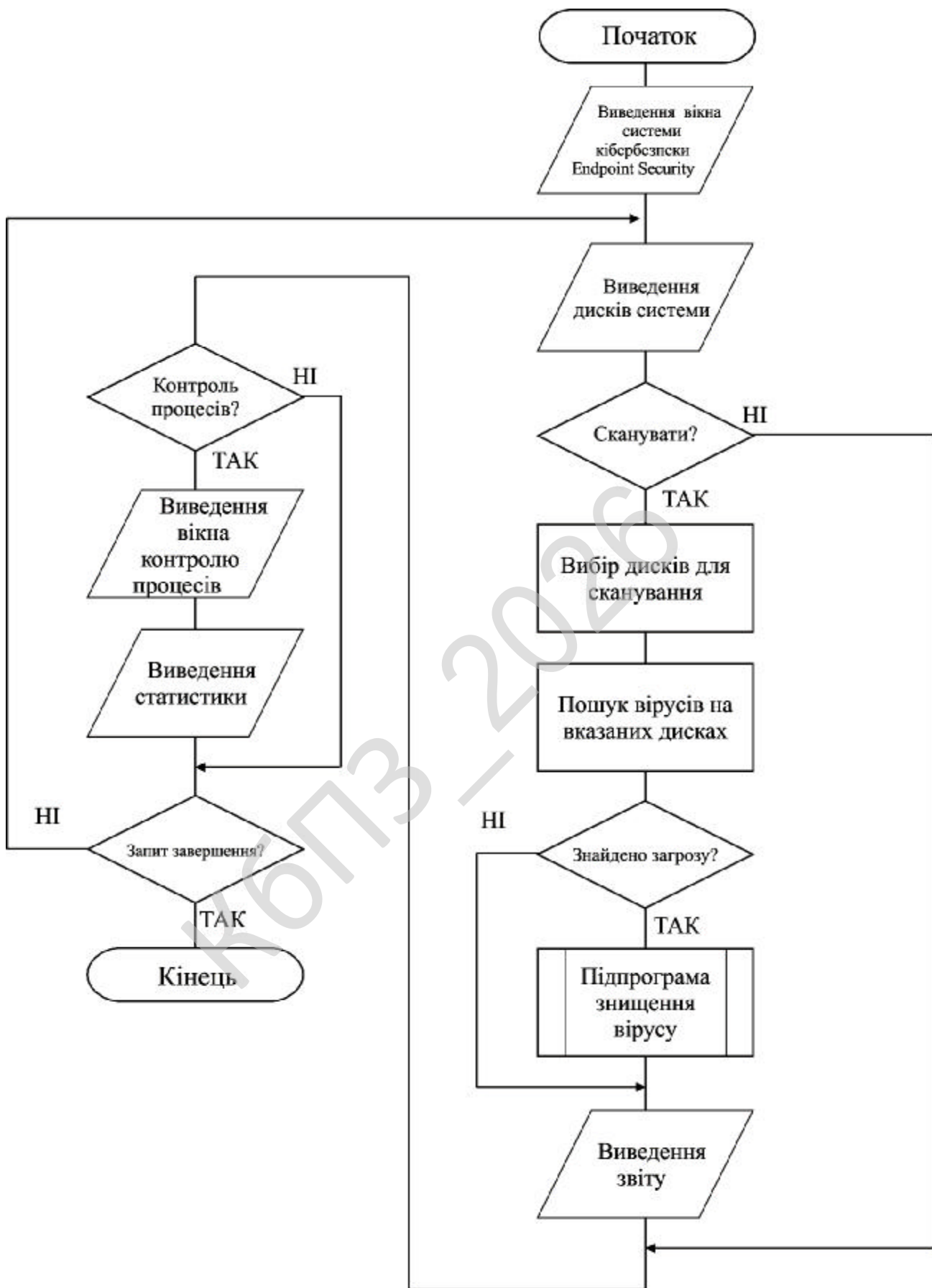


Рисунок 4.1 – Блок-схема основної програми

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

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.

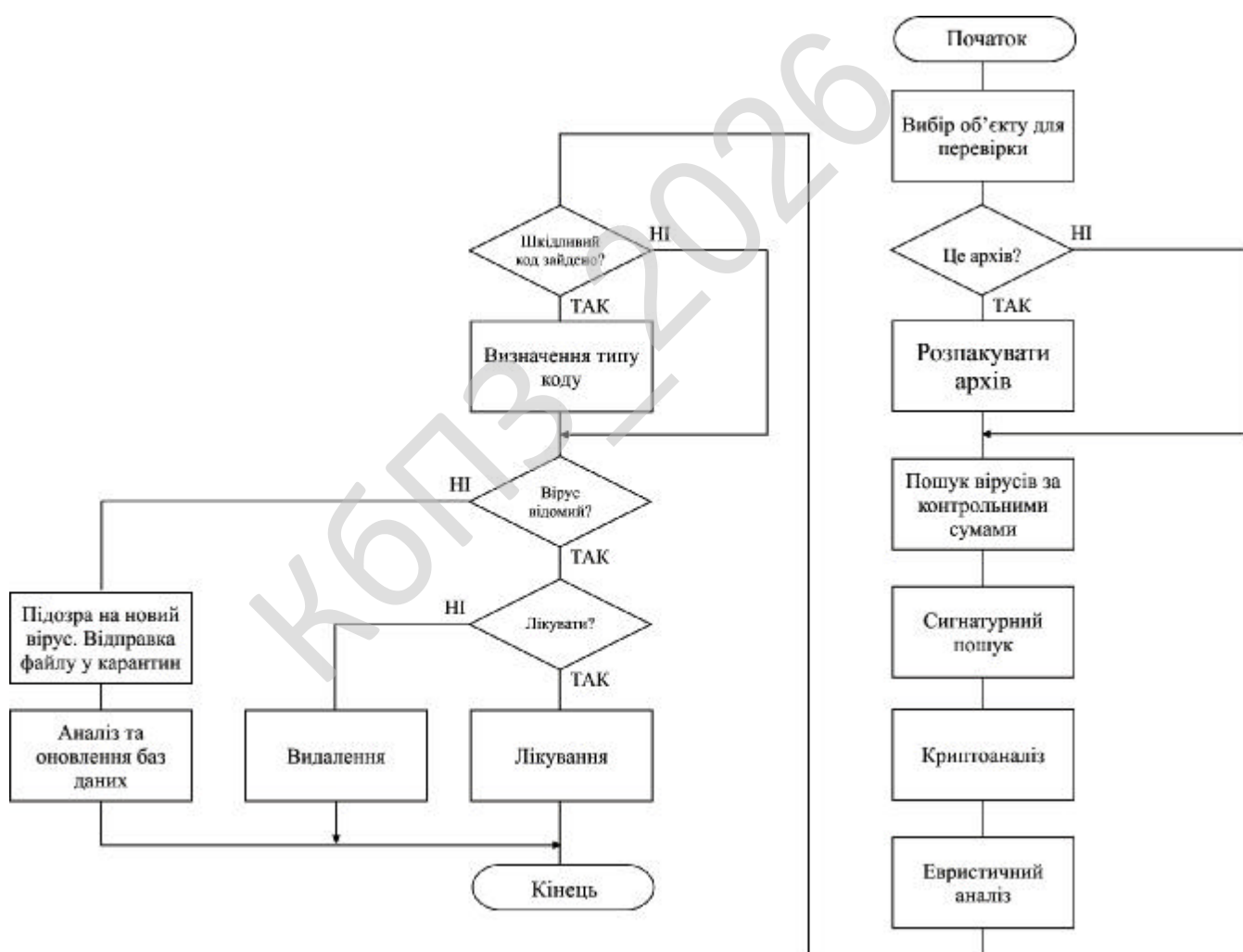


Рисунок 4.2 – Блок-схема роботи підпрограми

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:

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

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

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 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".

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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).

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.

Поліморфні віруси

Протягом всієї статті часто використовувалися терміни "поліморфні" віруси і віруси здатні до самошифрування. Саме цей тип шкідливих кодів вплинув на розвиток антивірусних технологій.

Самошифрування й поліморфічність використовуються практично всіма типами вірусів для того, щоб максимально ускладнити процедуру детектування вірусу. Поліморфні віруси (polymorphic) – це досить важко виявляемі віруси, які не мають сигнатур, тобто, не містять жодної постійної ділянки коду. У більшості випадків два зразки того самого поліморфного вірусу не будуть мати жодного збігу. Це досягається шифруванням основного тіла вірусу й модифікаціями програми-розшифровувача (декриптора). До поліморфних вірусів відносяться ті, детектування яких неможливо (або вкрай важко) здійснити за допомогою так званих вірусних масок – ділянок постійного коду, специфічних для конкретного вірусу. Досягається це двома основними способами – шифруванням основного коду вірусу з непостійним ключем і випадковим набором команд розшифровувача або зміною самого виконуваного коду вірусу. Існують також інші, досить екзотичні приклади поліморфізму: DOS-вірус "Bomber", наприклад, не зашифрований, однак послідовність команд, що передає керування коду вірусу, є повністю поліморфною.

Поліморфізм різного ступеня складності зустрічається у вірусах всіх типів – від завантажувальних і файлових DOS-Вірусів до Windows-вірусів і навіть макро-вірусів.

Поліморфні розшифровувачі

Найпростішим прикладом частково поліморфного розшифровувача є наступний набір команд, у результаті застосування якого жоден байт коду самого вірусу і його розшифровувача не є постійним при зараженні різних файлів:

MOV reg_1, count ; reg_1, reg_2, reg_3 вибираються з

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```

MOV reg_2, key ; AX, BX, CX, DX, SI, DI, BP
MOV reg_3, _offset ; count, key, _offset також можуть мінятися
_loop:
xxx byte ptr [reg_3], reg_2 ; xor, add або sub
DEC reg_1
Jxx _loop ; ja або jnc ; далі слідують зашифровані код і дані вірусу

```

Складні поліморфні віруси використовують значно більш складні алгоритми для генерації коду своїх розшифровувачів: наведені вище інструкції (або їхні еквіваленти) переставляються місцями від зараження до зараження, розбавляються нічого не змінюючими командами, типу NOP, STI, CLI, STC, CLC і т.д.

Повноцінні ж поліморфні віруси використовують ще більш складні алгоритми, у результаті роботи яких у розшифровувачі вірусу можуть зустрітися операції SUB, ADD, XOR, ROR, ROL і інші в довільній кількості й порядку. Завантаження й зміна ключів і інших параметрів шифровки виробляється також довільним набором операцій, у якому можуть зустрітися практично всі інструкції процесора Intel (ADD, SUB, TEST, XOR, OR, SHR, SHL, ROR, MOV, XCHG, JNZ, PUSH, POP ...) з усіма можливими режимами адресації. З'являються також поліморфні віруси, розшифровувач яких використовує інструкції аж до Intel386, а влітку 1997 року виявлений 32-бітний поліморфний вірус, що заражає EXE-Файли Windows 95.

У результаті, на початку файлу, зараженого подібним вірусом, іде набір безглузвих, на перший погляд, інструкцій. Цікаво, що деякі комбінації, які цілком працездатні, не беруться фірмовими дизасемблерами (наприклад, сполучення CS:CS: або CS:NOP). І серед цієї "каші" з команд і даних зрідка прослизують MOV, XOR, LOOP, JMP – інструкції, які дійсно є "робітниками".

Рівні поліморфізму

Існує розподіл поліморфних вірусів на рівні залежно від складності коду, що зустрічається в розшифровувачах цих вірусів. Такий розподіл уперше запропонував доктор Алан Соломон, через якийсь час Весселин Бончев розширив його:

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Рівень 1: Віруси, що мають деякий набір розшифровувачів з постійним кодом; при зараженні вибирають один з них. Такі віруси є "напів-поліморфними" і носять також назву "олігоморфних" (oligomorphic). Приклади: "Cheeba", "Slovakia", "Whale".

Рівень 2: Розшифровувач вірусу містить одну або кілька постійних інструкцій, основна ж його частина непостійна.

Рівень 3: Розшифровувач містить невикористовувані інструкції – "сміття" типу NOP, CLI, STI і т.д.

Рівень 4: У розшифровувачі використовуються взаємозамінні інструкції й зміна порядку проходження (перемішування) інструкцій. Алгоритм розшифровки при цьому не змінюється.

Рівень 5: Використовуються всі перераховані вище прийоми, алгоритм розшифровки непостійний, можливо повторне шифрування коду вірусу й навіть часткове шифрування самого коду розшифровувача.

Рівень 6: Permutating-Віруси. Зміні підлягає основний код вірусу – він ділиться на блоки, які при зараженні переставляються в довільному порядку. Вірус при цьому залишається працездатним. Подібні віруси можуть бути незашифрованні.

У наведеній вище класифікації є свої недоліки, оскільки вона виконується за єдиним критерієм – можливість детектувати вірус по коду розшифровувача за допомогою стандартного прийому вірусних масок:

Рівень 1: для детектування вірусу досить мати кілька масок.

Рівень 2: детектування по масці з використанням "wildcards".

Рівень 3: детектування по масці після видалення інструкцій-"сміття".

Рівень 4: маска містить кілька варіантів можливого коду, тобто стає алгоритмічною.

Рівень 5: неможливість детектування вірусу по масці.

Недостатність такого розподілу продемонстрована у вірусі 3-го рівня поліморфізму, що так і називається – "Level3". Цей вірус, будучи одним з

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

найбільш складних поліморфних вірусів, по наведеному вище розподілу попадає у рівень 3, оскільки має постійний алгоритм розшифровки, перед яким стоїть велика кількість команд-"сміття". Однак у цьому вірусі алгоритм генерування "сміття" доведений до досконалості: у коді розшифровувача можуть зустрітися практично всі інструкції процесора i8086.

Якщо зробити розподіл на рівні з погляду антивірусів, що використовують системи автоматичної розшифровки коду вірусу (емулятори), то розподіл на рівні буде залежати від складності емуляції коду вірусу. Можливо детектування вірусу й інших прийомів, наприклад, розшифровка за допомогою елементарних математичних законів і т.д.

Більш об'єктивною буде класифікація, у якій крім критерію вірусних масок беруть участь і інші параметри, наприклад:

- Ступінь складності поліморфного коду (відсоток від всіх інструкцій процесора, які можуть зустрітися в коді розшифровувача).
- Використання спеціальних прийомів, що утрудняють емуляцію антивірусами.
- Сталість алгоритму розшифровувача.
- Сталість довжини розшифровувача.

Зміна виконуваного коду

Найбільш часто подібний спосіб поліморфізму використовується макро-вірусами, які при створенні своїх нових копій випадковим чином міняють імена своїх змінних, вставляють порожні рядки або міняють свій код яким-небудь іншим способом. Таким чином, алгоритм роботи вірусу залишається без змін, але код вірусу практично повністю міняється від зараження до зараження.

Рідше цей спосіб застосовується складними завантажувальними вірусами. Такі віруси впроваджують у завантажувальні сектори лише досить коротку процедуру, що зчитує з диска основний код вірусу й передає на нього керування. Код цієї процедури вибирається з декількох різних варіантів (які також можуть бути розведені "порожніми" командами), команди переставляються

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

між собою й т.д.

Ще рідше цей прийом зустрічається у файлових вірусів – адже їм доводиться повністю міняти свій код, а для цього потрібні досить складні алгоритми. На сьогоднішній день відомі всього два таких віруси, один із яких ("Ply") випадковим чином переміщає свої команди по своєму тілу й заміняє їх на команди JMP або CALL. Інший вірус ("TMC") використовує більш складний спосіб – щоразу при зараженні вірус міняє місцями блоки свого коду й даних, вставляє "сміття", у своїх асемблерних інструкціях установлює нові значення оффсетів на дані, міняє константи й т.д. У результаті, хоча вірус і не шифрує свій код, він є поліморфним вірусом – у коді не присутній постійного набору команд. Більш того, при створенні своїх нових копій вірус міняє свою довжину.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики XP зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

Технологія екстремального програмування була розроблена Кентом Беком, Уардом Каннінгхемом та Роном Джеффріесом під час роботи над Chrysler

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Comprehensive Compensation System (C3). У 1996 Кент Бек став лідером проекту і почав вдосконалювати методи розробки, що застосовувалися в роботі над проектом. Свій метод він виклав у книзі «Extreme Programming Explained», котру було видано у жовтні 1999. Після купівлі Крайслера компанією Даймлер–Бенц проект C3 було скасовано у лютому 2000.

Хоча саме екстремальне програмування є відносно новим, багато її практик вже існували і використовувались протягом певного часу; однак, методологія підносить "найкращі практики" до екстремального рівня. Для прикладу, практика по плануванню і написанню тестів перед написанням кожної маленької частини коду було використано раніше в проекті НАСА "Меркурій". Для зменшення часу на розробку ПЗ деякі формальні документи тестування (такі як приймальне тестування) писались паралельно (або й раніше) з написанням самого ПЗ. Незалежна група тестування НАСА може писати процедури тестування базуючись на формальних вимогах до продукту до того як програмне забезпечення розроблене та інтегроване в систему. В ХР ця концепція піднесена до "екстремального рівня" завдяки написанню автоматичних тестів які перевіряють поведінку навіть малих частинок коду, а не тільки значних функціональних частин ПЗ.

Посібник Extreme Programming Explained: Embrace Change описує Екстремальне Програмування, як:

- Спроба примирити гуманність і продуктивність.
- Механізм для соціальної зміни.
- Шлях до удосконалення.
- Стиль розвитку.

Дисципліна розробки програмного забезпечення.

Головною метою Екстремального Програмування є скорочення вартості неочікуваних змін. У традиційних методах розробки (на кшталт SSADM) вимоги до розвитку системи визначаються на початку роботи над проектом, і часто виправляються пізніше. Це означає, що вартість проекту через зміни буде

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

більшою за заплановану (традиційна особливість для програмного забезпечення, що проектується).

ХР використовується для скорочення вартості змін, завдяки представленню простих значень, принципів і методів. При використанні екстремального програмування, проект повинен стати гнучкішим щодо змін.

Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше.

ХР намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає зміни до вимог як звичайні, неминучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваними. Основна ідея полягає в тому що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах – неминучі.

Екстремальне програмування також вводить набір практик та принципів на основі методології гнучкої розробки програмного забезпечення.

Екстремальне програмування описує чотири базові активності що виконуються при розробці програмного забезпечення: написання коду, тестування, слухання та дизайн.

Написання коду. Прихильники ХР заявляють що єдиним дійсно важливим результатом розробки ПЗ є код: без готового коду нема продукту.

Тестування. Методологія екстремального програмування заявляє, що якщо дрібне тестування може перевірити незначну частину функціональності, то багато дрібних тестів можуть перевірити набагато більше частинок і продукт в цілому.

Основні прийоми ХР. Дванадцять основних прийомів екстремального програмування (за першим виданням книги Extreme programming explained) можуть бути об'єднані в чотири групи:

1. Короткий цикл зворотного зв'язку (Fine scale feedback).
 - 1.1. Розробка через тестування (Test driven development).
 - 1.2 Гра в планування (Planning game).

- 1.3. Замовник завжди поруч (Whole team, Onsite customer).
- 1.4 Парне програмування (Pair programming).
- 2. Безперервний, а не пакетний процес.
- 2.1 Безперервна інтеграція (Continuous Integration).
- 2.2 Рефакторинг (Design Improvement, Refactor).
- 2.3 Часті невеликі релізи (Small Releases).
- 3. Розуміння, що поділяється всіма учасниками.
- 3.1 Простота (Simple design).
- 3.2 Метафора системи (System metaphor).
- 3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).
- 3.4 Стандарт кодування (Coding standard or Coding conventions).
- 4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

Опис функцій системи

Програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security працює як локальний агент захисту робочої станції. Система контролює стан кінцевого пристрою, перевіряє важливі файли, фіксує підозрілі процеси, аналізує мережеву активність і формує журнал подій безпеки.

Система реалізується мовою Python і орієнтується на захист даних користувача від несанкціонованого доступу, змінення, видалення або передавання за межі робочої станції. Основна логіка працює через модуль моніторингу, модуль аналізу, модуль реагування, базу подій і модуль звітності.

Архітектура системи має клієнтський принцип роботи. На комп'ютері користувача запускається агент Endpoint Security. Він збирає дані про файли, процеси та мережеві з'єднання. Після цього система порівнює поточний стан з правилами безпеки. Якщо виявляється ризик, подія зберігається в базі даних і відображається адміністратору.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Основні модулі системи наступні.

1. Модуль контролю файлів. Перевіряє цілісність важливих файлів за хеш сумою.
2. Модуль контролю процесів. Виявляє небажані або підозрілі процеси.
3. Модуль мережевого контролю. Фіксує активні мережеві з'єднання кінцевого пристрою.
4. Модуль правил безпеки. Визначає критерії небезпечної активності.
5. Модуль журналювання. Зберігає всі події безпеки в SQLite.
6. Модуль звітності. Формує короткий звіт для адміністратора.

Система виконує інвентаризацію контрольованих файлів. Вона обчислює хеш значення файлів і визначає факт їх зміни. Також система перевіряє список активних процесів і порівнює їх з переліком заборонених назв. Додатково агент аналізує мережеву активність і реєструє підозрілі підключення. Усі результати записуються до локальної бази даних.

Розглянемо частину коду що представляє як працює функціонал.

```
import os
import hashlib
import sqlite3
import psutil
from datetime import datetime

# Клас керує локальною базою подій безпеки
class SecurityDatabase:
    def __init__(self, db_path="endpoint_security.db"):
        self.db_path = db_path
        self.create_tables()

    def connect(self):
        return sqlite3.connect(self.db_path)

    def create_tables(self):
        with self.connect() as connection:
            cursor = connection.cursor()
            cursor.execute("""
                CREATE TABLE IF NOT EXISTS security_events (
                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                    event_time TEXT NOT NULL,
                    event_type TEXT NOT NULL,
                    object_name TEXT NOT NULL,
```

```

        description TEXT NOT NULL,
        risk_level TEXT NOT NULL
    )
    """
    connection.commit()
def add_event(self, event_type, object_name, description, risk_level):
    with self.connect() as connection:
        cursor = connection.cursor()
        cursor.execute("""
            INSERT INTO security_events
(event_time, event_type, object_name, description, risk_level)
VALUES (?, ?, ?, ?, ?)
            """, (
                datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                event_type,
                object_name,
                description,
                risk_level
            ))
        connection.commit()
def get_events(self):
    with self.connect() as connection:
        cursor = connection.cursor()
        cursor.execute("""
SELECT event_time, event_type, object_name, description, risk_level
FROM security_events
ORDER BY id DESC
            """)
        return cursor.fetchall()

# Клас перевіряє цілісність важливих файлів
class FileIntegrityMonitor:
    def __init__(self, database):
        self.database = database
        self.file_hashes = {}
    def calculate_hash(self, file_path):
        if not os.path.exists(file_path):
            return None
        hash_object = hashlib.sha256()
        with open(file_path, "rb") as file:
            for block in iter(lambda: file.read(4096), b''):
                hash_object.update(block)

```

```

        return hash_object.hexdigest()
def register_file(self, file_path):
    file_hash = self.calculate_hash(file_path)
    if file_hash is None:
        self.database.add_event(
            "FILE",
            file_path,
            "Файл не знайдено під час реєстрації",
            "MEDIUM"
        )
    return
    self.file_hashes[file_path] = file_hash
def check_files(self):
    for file_path, old_hash in self.file_hashes.items():
        current_hash = self.calculate_hash(file_path)
        if current_hash is None:
            self.database.add_event(
                "FILE",
                file_path,
                "Контрольований файл видалено",
                "HIGH"
            )
            continue
        if current_hash != old_hash:
            self.database.add_event(
                "FILE",
                file_path,
                "Виявлено зміну контрольованого файлу",
                "HIGH"
            )

# Клас аналізує активні процеси робочої станції
class ProcessMonitor:
    def __init__(self, database):
        self.database = database
        self.blocked_processes = {
            "keylogger.exe",
            "ransomware.exe",
            "unknown_stealer.exe"
        }
    def check_processes(self):
        for process in psutil.process_iter(["pid", "name"]):
            try:

```

```

        process_name = process.info["name"]
    if process_name and process_name.lower() in self.blocked_processes:
        self.database.add_event(
            "PROCESS",
            process_name,
            "Виявлено заборонений або підозрілий процес",
            "HIGH"
        )
    except (psutil.NoSuchProcess, psutil.AccessDenied):
        continue

# Клас контролює мережеву активність кінцевого пристрою
class NetworkMonitor:
    def __init__(self, database):
        self.database = database
        self.allowed_ports = {80, 443, 53}
    def check_connections(self):
        for connection in psutil.net_connections(kind="inet"):
            remote_address = connection.raddr
            if not remote_address:
                continue
            remote_port = remote_address.port
            if remote_port not in self.allowed_ports:
                object_name = f"{remote_address.ip}:{remote_port}"
                self.database.add_event(
                    "NETWORK",
                    object_name,
                    "Виявлено мережеве з'єднання через нетиповий порт",
                    "MEDIUM"
                )

# Клас об'єднує основні модулі Endpoint Security
class EndpointSecurityAgent:
    def __init__(self):
        self.database = SecurityDatabase()
        self.file_monitor = FileIntegrityMonitor(self.database)
        self.process_monitor = ProcessMonitor(self.database)
        self.network_monitor = NetworkMonitor(self.database)
    def prepare_demo_files(self):
        file_name = "important_data.txt"
        if not os.path.exists(file_name):
            with open(file_name, "w", encoding="utf-8") as file:
                file.write("Конфіденційні дані користувача")

```

```

        self.file_monitor.register_file(file_name)
def run_scan(self):
    self.file_monitor.check_files()
    self.process_monitor.check_processes()
    self.network_monitor.check_connections()
def print_report(self):
    events = self.database.get_events()
    print("Звіт Endpoint Security")
    print("Кількість подій", len(events))
    for event in events:
        event_time, event_type, object_name, description, risk_level = event
        print(f"{event_time} | {event_type} | {object_name} | {description} |
{risk_level}")
# Головна функція запускає демонстрацію роботи системи
def main():
    agent = EndpointSecurityAgent()
    agent.prepare_demo_files()
    agent.run_scan()
    agent.print_report()

```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використати фіналіста конкурсу AES – шифр Rijndael. Він є нетрадиційним блоковим шифром, оскільки не використовує мережу Фейштеля для криптоперетворень. Алгоритм представляє кожний блок кодуємих даних у вигляді двовимірної масиви байт розміром 4x4, 4x6 або 4x8 залежно від установленної довжини блоку. Далі на відповідних етапах перетворення відбуваються або над незалежними стовпцями, або над незалежними рядками, або взагалі над окремими байтами в таблиці. Всі перетворення в шифрі мають строге математичне обґрунтування. Сама структура й послідовність операцій дозволяють виконувати даний алгоритм ефективно як на 16-бітних так і на 64-бітних процесорах. У структурі алгоритму закладена можливість паралельного виконання деяких операцій, що на багатопроцесорних робочих станціях може ще підняти швидкість шифрування в 4 рази. Алгоритм складається з деякої кількості

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

раундів (від 10 до 14 – це залежить від розміру блоку й довжини ключа), у яких послідовно виконуються наступні операції :

ByteSub – Таблична підстановка 8х8 біт (рисунок 4.3).

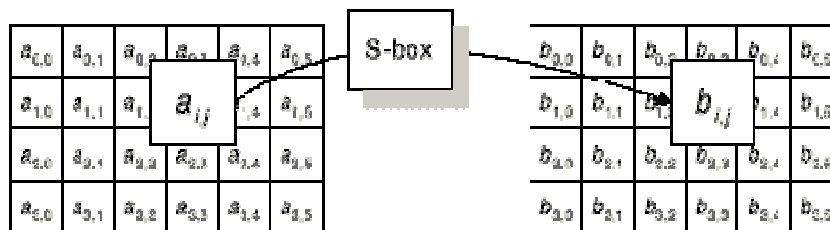


Рисунок 4.3 – Таблична підстановка 8х8 біт

ShiftRow – зрушення рядків у двовимірному масиві на різні зсуви (рисунок 4.4).

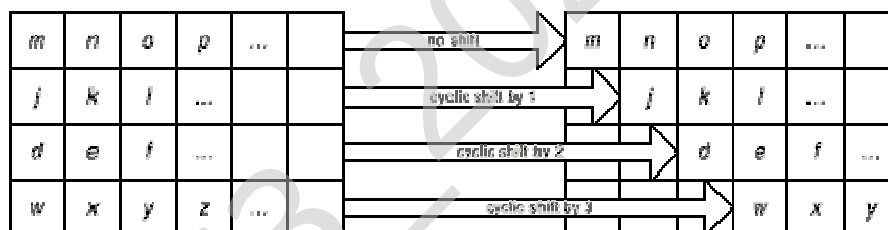


Рисунок 4.4 – Зрушення рядків у двовимірному масиві на різні зсуви

MixColumn – математичне перетворення, що перемішує дані усередині стовпця (рисунок 4.5).

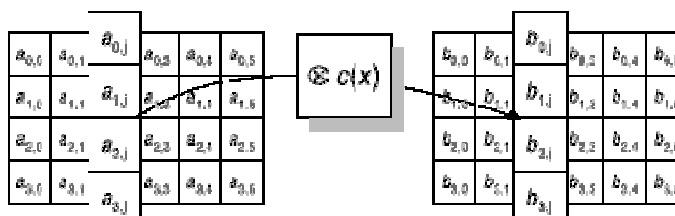


Рисунок 4.5 – Математичне перетворення, що перемішує дані усередині стовпця

AddRoundKey – додавання матеріалу ключа операцією XOR (рисунок 4.6).

$$\begin{array}{|c|c|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} & a_{0,5} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|c|c|} \hline k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} & k_{0,4} & k_{0,5} \\ \hline k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} & k_{1,5} \\ \hline k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} & k_{2,4} & k_{2,5} \\ \hline k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} & k_{3,4} & k_{3,5} \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} \\ \hline \end{array}$$

Рисунок 4.6 – Додавання матеріалу ключа операцією XOR

В останньому раунді операція перемішування стовпців відсутня, що робить всю послідовність операцій симетричною.

К6ПЗ_2026

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи кібербезпеки для захисту даних за рахунок використання Endpoint Security. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Розділу обрання функціональної вкладки.
- Розділу виведення результату роботи системи.

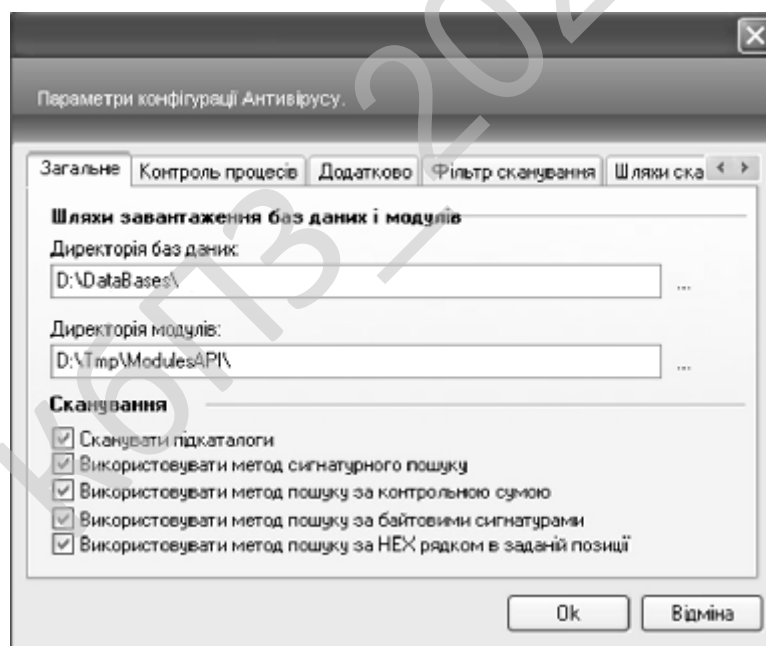


Рисунок 5.1 – Головне вікно ПЗ

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,

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

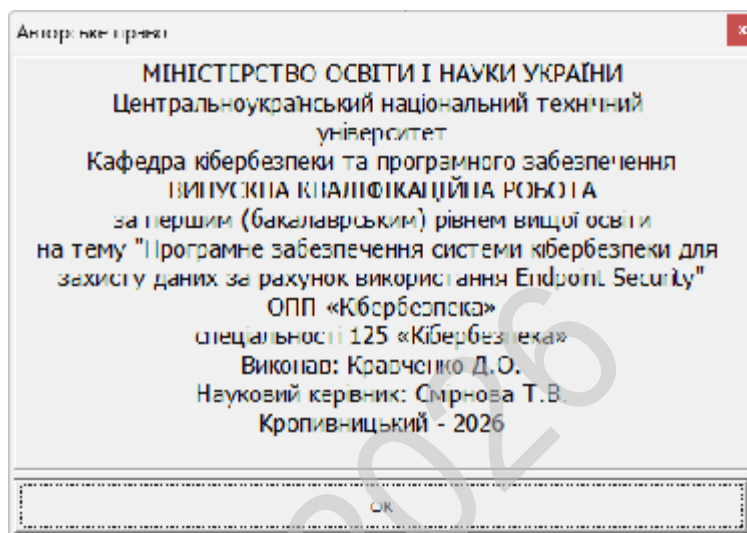


Рисунок 5.2 – Авторське право

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.

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

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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 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.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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.

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 (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 (zareestruvatisya), заплативши авторові певну суму.

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

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем для захисту даних за рахунок використання Endpoint Security.
- Досліджена система для захисту даних за рахунок використання Endpoint Security.
- На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для захисту даних за рахунок використання Endpoint Security.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для захисту даних за рахунок використання Endpoint Security.

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 системи кібербезпеки для захисту даних за рахунок використання Endpoint

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Security. 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 кібербезпеки 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 AES.

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-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vijay Kumar Velu. Mastering Kali Linux for Advanced Penetration Testing. Packt Publishing Ltd. 2022. 573 p.
2. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 p.
3. Massimo Bertaccini. Cryptography Algorithms. Packt Publishing. 2022. 358 p.
4. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.
5. 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.
6. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
7. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
8. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
9. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
10. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
11. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
12. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
13. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications*, 2026, pp. 587-633.
14. 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.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

15. Chulinda L., Smirnov O., Shapenko L., Ustynova I., Bohatiuk I., Kelyp S. «The role of innovation in ensuring the safety of international civil aviation». *Ceur Workshop Proceedings*, 2025, 4024, pp. 530–542.

16. 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.

17. 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

18. 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.

19. 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.

20. 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.

21. Lakhno, V., Malyukov, V., Smirnov, O., Bebesko, 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

29. 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.

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

31. 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

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

33. 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.

34. 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, KL., Vuppapapati, 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.

35. 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.

36. 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 (ІПШРІТ-2023))» м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

37. 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.

38. 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.

39. 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.

40. 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,

41. 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.

42. 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.

					ВКРБ-125.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

43. 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.

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

45. 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.

46. 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

47. 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.

48. 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.

49. 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.

50. 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.

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

52. 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.