

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
корпоративної мережі з застосуванням SPI Firewall”

Виконав здобувач вищої освіти
IV курсу, групи КБ-22-2
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Спринчан Є.С.
«___» _____ 2026 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
«___» _____ 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
« 15 » січня 2026 року

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

Спринчан Єлизаветі Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall*

2. Керівник роботи *Смірнов Олексій Анатолійович, доктор технічних наук, професор*

(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 корпоративної мережі з застосуванням SPI Firewall*

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 « 15 » січня 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
« 15 » січня 2026 р.

Підпис керівника

Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 15 » січня 2026 р.

Підпис здобувача

Спринчан Є.С.
(прізвище та ініціали)

АНОТАЦІЯ

Спринчан Є.С. Програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall. 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 корпоративної мережі з застосуванням SPI Firewall.

Метою розробки є програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

Результат роботи – програмна реалізація системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

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 Visual C++.

Ключові слова: кібербезпека, SPI Firewall

ABSTRACT

Sprynchan E.S. Software for a cybersecurity system for corporate network protection using SPI Firewall. 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 corporate network protection using SPI Firewall.

The purpose of the development is software for a cybersecurity system for corporate network protection using SPI Firewall.

The result of the work is a software implementation of a cybersecurity system for corporate network protection using SPI Firewall.

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

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

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

The program was developed in Visual C++.

Keywords: cybersecurity, SPI Firewall

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	15
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	28
3.4 Розробка діаграми процесів.....	33
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	35
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	35
4.2 Захист розробленого програмного забезпечення.....	50
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	53
6 ОСНОВНІ ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62

					ВКРБ-125.26.0041.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	Спринчан Є.С.					Б	1	69
<i>Перев.</i>	Смірнов О.А.							
<i>Н.контр.</i>	Коваленко А.С.					<i>ЦНТУ КБ-22-2</i>		
<i>Затв.</i>	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЛОМ	—	локальна обчислювальна мережа
MME	—	міжмережні екрани
ATM	—	асинхронний режим передачі
BSD	—	адаптована для Internet реалізація операційної системи UNIX
ICMP	—	міжмережний протокол управляючих повідомлень
IP	—	Internet Protocol – міжмережний протокол
NFS	—	мережна файлова система
PPP	—	протокол передачі від точки до точки
RFC	—	опис набору протоколів Internet
RPC	—	віддалений виклик процедури
SLIP	—	міжмережний протокол для послідовного каналу
SMTP	—	Simple Mail Transfer Protocol – простий протокол передачі пошти
TCP	—	Transmission Control Protocol – протокол управління передачею
UDP	—	User Datagram Protocol – протокол користувальницьких датаграм
UNIX	—	багатозадачна операційна система
UTP	—	незахищена вита пара
URL	—	уніфікований покажчик інформаційного ресурсу

ВСТУП

Актуальність теми. Програма брандмауера, розгорнута на одній кінцевій точці, такий як ПК або сервер, функціонує як захист для цієї системи. Вона перевіряє вхідний та вихідний трафік, застосовуючи набори правил для блокування шкідливої активності, обмеження поведінки програм та запобігання несанкціонованій передачі даних. На відміну від апаратних брандмауерів, які захищають цілі мережі, програмні брандмауери працюють на рівні пристрою на основі вбудованих функцій, призначених для захисту системи від шкідливих атак або загроз.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

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

- Огляд існуючих систем для захисту корпоративної мережі з застосуванням SPI Firewall.
- Дослідження системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.
- Програмна реалізація системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для захисту корпоративної мережі з застосуванням SPI Firewall.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Призначення міжмережевого екрана

Міжмережевий або мережний екран являє собою комплекс устаткування й програмних засобів, яким контролюються й фільтруються минаючі мережні пакети, з огляду на попередньо встановлені критерії. Цей захисний екран, називаний брандмауером або Firewall, захищає від атаки хакерів локальні корпоративні мережі, запобігає несанкціонованому зовнішньому доступу, спробам зламати сховища даних і зробити крадіжку інформації. Корпоративні користувачі, завдяки брандмауерові, одержують безпечний доступ до світової павутини, контрольоване віддалене підключення зовнішніх пристроїв до локального IT-інфраструктурі.

Firewall виконує два основні завдання:

- захист корпоративної комунікаційної системи (комп'ютера) від зовнішнього проникнення;
- запобігання несанкціонованій вихідній передачі даних.

Компанії, вибираючи для власних бізнес-завдань певний міжмережевий екран, зобов'язані керуватися законодавством, що забезпечує стандарти роботи з персональними даними. У ряді випадків, а саме при роботі з інформацією обмеженого доступу, обов'язковим є застосування брандмауера.

SPI Firewall – ключовий елемент захисту

Завдяки своїй гнучкості, простоті адміністрування, модульній конструкції, SPI Firewall активно застосовується для створення діючої системи безпеки, що охороняє комунікаційні мережі.

Блокуючи спроби підключитися до корпоративної мережі ззовні, Firewall пропускає запити вихідного характеру. При роботі із вхідними з'єднаннями, SPI

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Firewall автоматично не перешкоджає запитам, що не порушують установлені конфігурації безпеки, створені поштовим шлюзом або проксі-сервером. Фільтрація вступників запитів на з'єднання здійснюється з урахуванням оцінки небезпеки по різних параметрах, а саме:

- мережам і IP-адресам;
- використовуваним портам і протоколам.

Серед основних переваг SPI Firewall експерти виділяють:

– захист мережі від вірусних погроз, що дозволяє убезпечити бізнес в Інтернеті;

– завдяки можливості здійснювати фільтрацію контенту, блокування сайтів, створюючи власний обмежувальний список, підвищується продуктивність праці співробітників, що позначається на ефективності й прибутковості бізнесу;

– можливість працювати з маршрутизаторами, використовуючи інтегровані сервіси, що дозволяє домогтися інтелектуальної маршрутизації й оптимального розподілу навантаження;

- простоту адміністрування й налаштування.

Основні переваги використання:

– Допомагають захищати бізнес від Інтернет-хробаків, крадіжки даних і мережних атак.

– Збільшують вартість підприємства шляхом підвищення ефективності й продуктивності.

– Працюють у сполученні з маршрутизаторами з інтегрованими сервісами.

– Прості у використанні й керуванні.

1.2 Область застосування

Брандмауер роутера – це базовий бар'єр проти кіберзагроз, але його можливостей часто недостатньо. Хоча він фільтрує вхідні атаки, у нього є дві вразливості: він «не бачить» підозрілу активність програм, які вже потрапили на

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

пристрій і відправляють ваші дані назовні, а також стає безсилим, коли ви підключаєтеся до публічних Wi-Fi мереж.

Для надійного захисту варто встановити якісний антивірус із вбудованим фаєрволом безпосередньо на ваш гаджет – це забезпечить контроль трафіку в будь-якому місці та захистить від внутрішніх загроз.

Коли ми додаємо програмні брандмауери до наших пристроїв у домашніх мережах, це робиться не лише для захисту пристроїв, які ми використовуємо. Розумний брандмауер гарантує, що якщо один пристрій заразиться, він не передасть вірус іншим пристроям у нашій мережі.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

Найкраще програмне забезпечення для брандмауерів 2026 рік

1. Bitdefender Total Security

Найкращий захист для всебічної безпеки завдяки захисту брандмауером

Bitdefender Total Security – це потужний пакет інтернет-безпеки, який пропонує брандмауер серед низки функцій захисту. Також є антивірусний захист, але його багаторівневий захист від шкідливого програмного забезпечення також захищає від програм-вимагачів.

Інші функції включають опції захисту від фішингу, шахрайства та крадіжки, а також оптимізатор продуктивності. Широкий спектр опцій захисту конфіденційності включає захист від відстеження, знищення файлів, батьківський контроль, а також захист мікрофонів і веб-камер, щоб запобігти їх використанню сторонніми програмами без вашої згоди. Він також пропонує вбудований VPN з розумним лімітом 200 МБ/день/пристрій.

Bitdefender Total Security призначений не лише для захисту користувачів Windows, але й може використовуватися для захисту від атак на Android, macOS та iOS. Ще краще те, що одна ліцензія на продукт дозволяє використовувати програмне забезпечення на 5 різних пристроях, тож ви можете використовувати Bitdefender Total Security як для домашніх ПК, так і для смартфонів.

Загалом, Bitdefender Total Security практично виконує свою назву, пропонуючи дуже широкий спектр засобів безпеки, що виходять за рамки простого брандмауера. Він коштує \$59,99 при оплаті на рік або \$129,99 при оплаті на два роки.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Захистіть свої пристрої за допомогою наших купонів Bitdefender. Отримайте першокласну кібербезпеку за зниженою ціною та забезпечте безпеку своєї онлайн-діяльності.

2. Norton 360 Deluxe

Найкращий багатофункціональний захист брандмауера

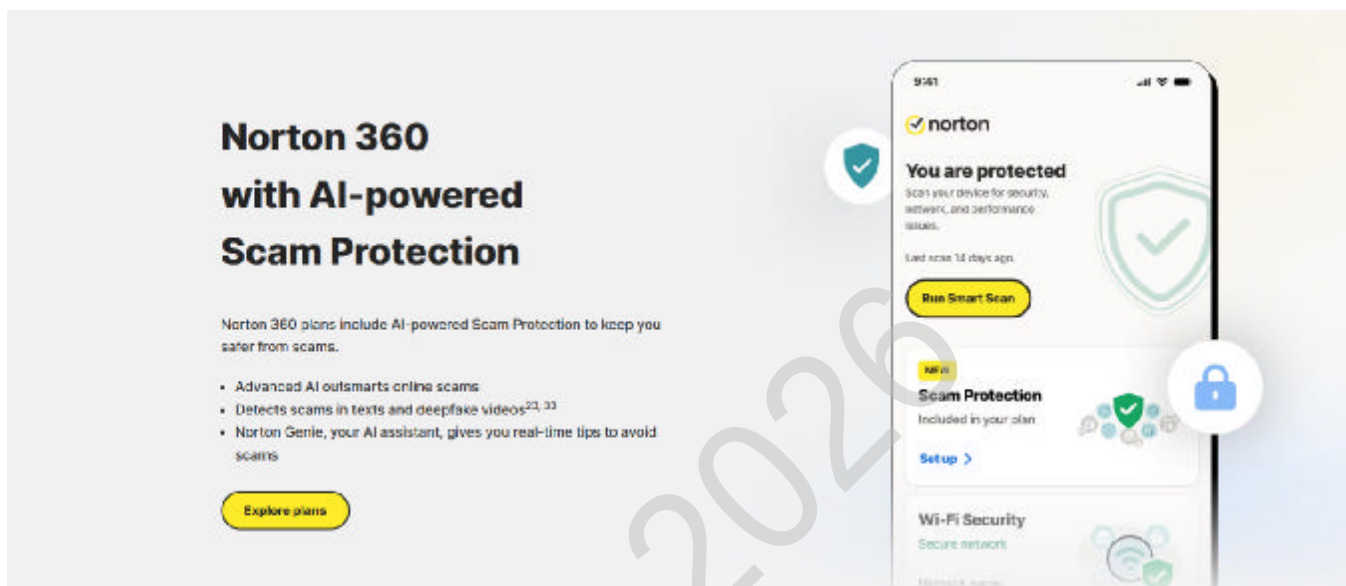


Рисунок 2.1 – Інтерфейс користувача Norton 360 Deluxe

Norton 360 Deluxe пропонує ще багатший та функціональніший пакет інтернет-безпеки, ніж просто брандмауер. Хоча він забезпечує антивірусний та стандартний захист від шкідливого програмного забезпечення, він також пропонує ще більше можливостей.

Додаткові функції Norton 360 Deluxe включають безпечний VPN, батьківський контроль, менеджер паролів (хоча ви можете використовувати найкращий менеджер паролів) та онлайн-резервне копіювання в хмару. Також є функція моніторингу даркнету, яка сповіщатиме вас про виставлення ваших даних на продаж у даркнеті, та захист від шахрайства на основі штучного інтелекту, щоб захистити вас та ваш банківський рахунок від шахраїв.

Norton 360 Deluxe коштує \$49,99 за перший рік, але потім його можна

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

продовжити за \$119,99 на рік, що робить його дорожчим, ніж деякі інші перелічені тут рішення, такі як Bitdefender Total Security. Але, як і Bitdefender Total Security, Norton 360 Deluxe працює на Windows, macOS, iOS та Android, а одна ліцензія покриває до 5 пристроїв загалом.

Якщо ви хочете додати захист від крадіжки особистих даних, ви можете оновити Norton 360 до Lifelock, який забезпечує додатковий захист особистих даних та моніторинг кредитної історії, а також страхове положення у разі неправомірного використання вашої особистої інформації.

3. Avast Premium Security

Найкращий варіант брандмауера для кількох пристроїв



Рисунок 2.2 – Інтерфейс користувача Avast Premium Security

Avast Premium Security – це ще один платний повнофункціональний пакет інтернет-безпеки, який можна придбати для одного пристрою, будь то Windows,

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

macOS, iOS або Android, або як пакет, що охоплює до 10 пристроїв, незалежно від платформи.

Як і у випадку з Bitdefender та Norton, це означає, що ви можете захистити як домашній ПК, так і мобільні пристрої. Однак, оскільки є можливість захистити до 10 пристроїв, це означає, що ви можете використовувати одну ліцензію також для захисту родини.

Окрім брандмауера для всіх пристроїв, Avast Premium Security також забезпечує захист від програм-вимагачів та опції знищення файлів. Також є захист від фішингу та шахрайства для безпечнішого онлайн-банкінгу, а також захист від несанкціонованого використання вашої веб-камери.

Загалом, Avast Premium Security – це ще один надійний пакет інтернет-безпеки, але одна річ, яка його підводить, – це постійні спливаючі вікна з рекламою, що пропонують перехресні продажі програмного забезпечення для налаштування ПК та іншого програмного забезпечення, з чим людям, які заплатили за ліцензію, не доведеться постійно стикатися.

Також, на відміну від деяких своїх аналогів, Avast Premium Security не включає VPN, хоча він заплутано відображається на панелі інструментів, але вимагає окремої підписки.

Вартість Avast Premium Security для одного пристрою становить 49,08 доларів США на рік протягом першого року, а поновлення – 77,99 доларів США на рік. Для 10 пристроїв вартість першого року – 69,48 доларів США на рік, а наступні роки – 99,99 доларів США на рік.

4. Panda Dome Essential

Найкраще співвідношення ціни та якості брандмауера та рішення для інтернет-безпеки.

Брандмауер, виявлення шкідливого програмного забезпечення в режимі реального часу, сканування USB, захист Wi-Fi, захист від фішингу: Panda Dome Essential має всі необхідні функції – не кажучи вже про цінні додаткові функції – яких ви очікуєте від найкращого антивірусного програмного забезпечення.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

І це лише початок. Panda Dome сповнений додаткових функцій, з якими не можуть зрівнятися деякі інші плани у цьому списку, зокрема сканер даркнету та вбудований VPN. Однак йому також бракує деяких функцій, які ви отримаєте з іншими продуктами в цьому посібнику. Наприклад, немає батьківського контролю чи захисту від програм-вимагачів, які ви отримуєте з іншими.

Як і його аналоги, Panda Dome Essential захищає не лише ПК з Windows, але й macOS, а також смартфони та планшети Android та iOS.

Ще однією безсумнівною перевагою є ціна – Panda Dome Essential є однією з найкращих програм за співвідношенням ціни та якості. І у вас навіть є можливість заплатити трохи більше та оновити підписку, щоб охопити необмежену кількість ноутбуків, телефонів, планшетів та настільних комп'ютерів за допомогою однієї підписки.

Вартість Panda Dome Essential для одного пристрою становить 34,99 дол. США на рік або 69,99 дол. США за два роки. Ви можете охопити до 3 пристроїв за 41,99 дол. США на рік, 5 пристроїв за 48,99 дол. США, 10 пристроїв за 69,99 дол. США або будь-яку кількість пристроїв за 83,99 дол. США.

5. Webroot SecureAnywhere

Майже кожен антивірусний інструмент претендує на звання «легкого», але Webroot Essentials – єдиний, хто справді відповідає цим вимогам. Він має мінімальний вплив на системні ресурси, що робить його гарним варіантом для старіших або менш потужних комп'ютерів.

Webroot забезпечує захист від широкого спектру загроз, включаючи віруси, шкідливе програмне забезпечення, програми-вимагачі та фішингові атаки. Він також включає такі функції, як брандмауер і менеджер паролів. Але, як і Panda Dome Essential, Webroot Essential також не має VPN та батьківського контролю.

Нелегко порівняти точність Webroot з конкурентами, оскільки великі тестові лабораторії рідко оцінюють продукти компанії. Однак, коли їх перевіряють, вони зазвичай отримують високі бали, а наші власні тести

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

показують надійний та надійний захист.

Ще однією перевагою є те, що продукт підтримує комп'ютери Windows та macOS, а також смартфони та планшети Android та iOS. Ви можете придбати Webroot Essentials для захисту одного пристрою, 3 пристроїв або 5 пристроїв.

Щоб захистити 1 пристрій, вам доведеться заплатити 37,49 долара за 1 рік, 43,74 долара за 2 роки або 45,83 долара за 3 роки. Аналогічно, для 3 пристроїв це коштуватиме 52,49 долара за 1 рік, 61,24 долара за 2 роки та 64,16 долара за 3 роки. План захисту для 5 пристроїв коштує 67,49 долара за 1 рік, 78,74 долара за 2 роки та 82,49 долара за 3 роки.

У Webroot Essentials є багато переваг, а 70-денна гарантія повернення грошей зі 100% гарантією свідчить про те, що компанія впевнена в продукті. Якщо вам набридли надто складні та роздуті движки, Webroot – гідний претендент для вас.

Найкращий безкоштовний брандмауер 2026 року

1. ZoneAlarm Free Antivirus

ZoneAlarm Free Antivirus, також відомий як ZoneAlarm Free Antivirus + Firewall, має на меті забезпечити більше, ніж просто захист від шкідливих програм. Позиціонований як легкий пакет безпеки для користувачів Windows, він поєднує класичні антивірусні інструменти з функціональністю брандмауера та кількома додатковими функціями.

По суті, ZoneAlarm пропонує антивірусний та антишпигунський захист, призначений для виявлення широкого спектру загроз, включаючи віруси, боти, черв'яки та трояни. Він пропонує два типи сканування: критичне сканування, яке швидше та сканує лише критичні області файлової системи та ОС, і повне сканування, яке переглядає всю файлову систему на локальному диску, а також на будь-яких підключених USB-пристроях.

Він також має двосторонній брандмауер, який контролює трафік, що входить і виходить з вашої мережі, робить ваш ПК невидимим для хакерів і запобігає поширенню ваших особистих даних шпигунськими програмами.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

ZoneAlarm доступний як у безкоштовній, так і в професійній версіях. У першій відсутні деякі функції, зокрема експертні правила, захист від фішингу, веб-монітор, емуляція загроз тощо. Вона виявлятиме та блокуватиме потенційно небезпечний трафік, приховуватиме відкриті порти та дозволить блокувати підключення будь-якої програми до Інтернету.

2. TinyWall

Найкраще, якщо вам потрібен ненав'язливий безкоштовний брандмауер

TinyWall насправді розроблений для використання з вбудованим брандмауером Defender у Windows, тому його вплив на продуктивність мінімальний.

TinyWall активно блокуватиме трояни, віруси, черв'яків та інші форми шкідливого програмного забезпечення, а також запобігатиме зміні налаштувань брандмауера Windows шкідливими програмами. І все це без необхідності знати про порти, протоколи чи деталі програм.

Але, мабуть, найпривабливішою його особливістю є відсутність спливаючих вікон. Більшість брандмауерів відображатимуть спливаюче вікно із запитанням про те, що ви хочете зробити, щоразу, коли програма намагається отримати доступ до Інтернету. Це не тільки дратує, але й зрештою ви просто почнете надавати доступ до всього, щоб позбутися спливаючих вікон, що не дуже безпечно.

TinyWall обходить цю проблему, очікуючи, що ви додасте до білого списку програми, яким ви дійсно хочете надати доступ до Інтернету. Коли ви виявите, що програму заблоковано, її легко додати до білого списку, а потім ви можете продовжити свої дії.

3. Захисник Microsoft

Найкращий варіант, якщо ви не хочете нічого встановлювати.

Якщо у вас ПК з версією Windows, новішою за XP, то у вас вже встановлено безкоштовний брандмауер, і якщо ви не змінили жодних налаштувань, він уже активний і працює.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Відомий як Захисник Windows з моменту випуску Windows 10, брандмауер є частиною набору інструментів безпеки, що входять до комплекту інсталяцій Windows за замовчуванням. Microsoft рекомендує залишати Захисник увімкненим, навіть якщо ви також використовуєте інший брандмауер.

Щоб керувати його налаштуваннями, просто знайдіть «безпека» у Windows і виберіть програму «Безпека». Коли вона відкриється, увімкніть брандмауер Захисника Windows. Щоб вимкнути його знову, просто натисніть ту саму кнопку.

Ви також можете налаштувати брандмауер так, щоб він пропускав певну програму через нього. Там само, де ви його вмикаєте/вимикаєте, ви можете вибрати «Дозволити програмі проходити через брандмауер», а потім вибрати потрібну програму.

4. GlassWire

GlassWire – це інструмент для моніторингу мережі та роботи з брандмауером, відомий своїм продуманим інтерфейсом та зручним дизайном. Його безкоштовна версія – гарний варіант для тих, хто хоче краще контролювати свою інтернет-активність, не потребуючи звикання до складних налаштувань брандмауера.

Одна з ключових функцій безкоштовної версії – це можливість візуально відстежувати мережеву активність у режимі реального часу та в історії. Ви можете бачити, які програми використовують пропускну здатність, до яких серверів вони підключаються та скільки даних передається. Ця інформація може допомогти вам ідентифікувати програми, які використовують багато пропускну здатності, або навіть виявити неочікувану поведінку мережі.

Що стосується функціональності брандмауера, безкоштовна версія містить базову функцію блокування одним кліком, яка дозволяє блокувати доступ до Інтернету для двох програм. Однак багато розширених функцій доступні лише у преміум-версії за \$2,99/місяць. Наприклад, безкоштовна версія не включає можливість виявлення нових або незнайомих пристроїв у вашій локальній

мережі, а також не дозволяє створювати кілька профілів брандмауера для різних мереж.

Ось чому найкраще використовувати GlassWire Free разом із брандмауером Windows, використовуючи його радше як монітор мережевого трафіку, ніж як брандмауер.

Як вибрати найкращий брандмауер

Вибираючи, який брандмауер завантажити та використовувати, спочатку врахуйте свої фактичні потреби в безпеці. Іноді безкоштовне та бюджетне програмне забезпечення може пропонувати лише базові опції, тому, якщо вам потрібно використовувати розширені інструменти, платна платформа може бути набагато вигіднішою. Крім того, пакети програмного забезпечення для безпеки вищого класу зазвичай можуть задовольнити будь-які потреби, наприклад, надати інші інструменти, такі як вбудований VPN, антивірус та захист від шкідливих програм, тому переконайтеся, що ви добре розумієте, які функції, на вашу думку, можуть знадобитися від вашого брандмауера.

Також важливо враховувати відмінності між VPN та брандмауером, вибираючи найкращий сервіс для ваших потреб.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програми мною була використана мова програмування Visual C++. У зв'язку з тим, що сьогодні рівень складності програмного забезпечення дуже високий, розробка застосунків Windows з використанням тільки якої-небудь мови програмування значно утрудняється. Програміст повинен затратити масу часу на рішення стандартних завдань по створенню багатовіконного інтерфейсу. Реалізація технології зв'язування й вбудовування об'єктів – OLE – зажадає від програміста ще більш складної роботи. Щоб полегшити роботу програміста практично всі сучасні компілятори з мови C++

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

містять спеціальні бібліотеки класів. Такі бібліотеки містять у собі практично весь програмний інтерфейс Windows і дозволяють користуватися при програмуванні засобами більш високого рівня, чим звичайні виклики функцій. За рахунок цього значно спрощується розробка застосунків, що мають складний інтерфейс користувача, полегшується підтримка технології OLE і взаємодія з базами даних. Сучасні інтегровані засоби розробки застосунків Windows дозволяють автоматизувати процес створення застосунка. Для цього використовуються генератори застосунків. Програміст відповідає на питання генератора застосунків і визначає властивості застосунка – чи підтримує воно багатовіконний режим, технологію OLE, тривимірні органи керування, довідкову систему. Генератор застосунків, створить додаток, що відповідає вимогам, і надасть вихідні тексти. Користуючись їм як шаблоном, програміст зможе швидко розробляти свої застосунки. Подібні засоби автоматизованого створення застосунків включені в компілятор Microsoft Visual C++ і називаються MFC AppWizard. Заповнивши кілька діалогових панелей, можна вказати характеристики застосунка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні застосунки, а також застосунки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину застосунка програмістові прийдеться розробляти самостійно. Вихідний текст застосунка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон застосунка – це вже половина всієї роботи. Вихідні тексти застосунків, автоматично отриманих від MFC AppWizard, можуть становити сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти застосунків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої застосунки

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-застосунки на базі об'єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-застосунків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованого програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API) Windows. Будь-який додаток взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Оскільки MFC являє собою набір класів, написаних мовою C++, тому програми, написані з використанням MFC, повинні бути в той же час програмами на C++. Для цього необхідно володіти відповідними знаннями. Для початку необхідно вміти створювати власні класи, розуміти принципи спадкування й вміти перевизначати віртуальні функції. Хоча програми, що використовують бібліотеку MFC, звичайно не містять занадто специфічних елементів з арсеналу C++, для їхнього написання проте потрібні солідні знання в даній області.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

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

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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.

К6ПЗ-2026

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Найважливіші функції програмного забезпечення брандмауера:

- Технології ідентифікації.
- Перевірка вмісту.
- VPN.
- Проксі-сервер.
- Державна інспекція.
- Консоль керування брандмауером.
- Контроль на основі політик.
- Висока доступність.

Технології ідентифікації

У програмному забезпеченні брандмауерів технології ідентифікації розширюють видимість за межі IP-адрес, пов'язуючи трафік з окремими користувачами та пристроями. Ця функціональність дозволяє адміністраторам застосовувати детальні політики на основі ролей та налаштовувати засоби контролю доступу. Таке застосування на основі ідентифікації є визначальною рисою брандмауерів наступного покоління (NGFW).

Переваги:

- Завдяки інтеграції ідентифікації користувача, брандмауери посилюють безпеку за допомогою багатофакторної автентифікації та проактивного виявлення ризикованої поведінки, такої як надсилання облікових даних на неперевірені сайти.
- Зменшує загрози, пов'язані з обліковими даними, забезпечуючи при цьому неперевершену видимість мережевої активності.
- Допомогає організаціям отримати можливість відстежувати моделі

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

використання, оцінювати споживання пропускної здатності програмами та приписувати події безпеки окремим користувачам, що забезпечує швидше та точніше реагування на інциденти та проведення судово-медичних розслідувань.

Перевірка вмісту

Технології перевірки вмісту брандмауера виходять за рамки базової фільтрації пакетів, аналізуючи фактичні дані в мережевому трафіку. Такі методи, як глибока перевірка пакетів, перевірка стану, проксування та контроль програм, у поєднанні із сучасними функціями NGFW, такими як штучний інтелект/машинне навчання, дозволяють виявляти шкідливе програмне забезпечення, експлойти та конфіденційні дані, приховані в трафіку, що виглядає легітимним.

Переваги:

- Брандмауери можуть перехоплювати та перевіряти трафік до його проходження, забезпечуючи ретельну перевірку контенту.
- Виявляйте складні та невідомі загрози, аналізуючи поведінку трафіку, а не покладаючись виключно на сигнатури.
- Контролюйте доступ до ризикованого або невідповідного контенту, блокуючи сайти, файли або категорії відповідно до корпоративної політики.

VPN

VPN, інтегрований з технологією брандмауера, забезпечує багаторівневий захист. Брандмауер діє як гейткіпер, блокуючи небажаний трафік, тоді як VPN шифрує потоки даних та приховує IP-адреси для забезпечення конфіденційності. Таке поєднання забезпечує безпечні тунелі для віддаленого доступу, гарантуючи, що проходить лише автентифікований, зашифрований трафік. Апаратні засоби та програмні рішення забезпечують організаціям стійкий захист від шкідливих програм та несанкціонованого вторгнення.

Переваги:

- Дозволяє співробітникам безпечно підключатися до ресурсів компанії з будь-якого місця, захищаючи конфіденційні дані.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- Шифрує весь інтернет-трафік для забезпечення приватності та конфіденційності перегляду.
- Блокує шкідливе програмне забезпечення та спроби несанкціонованого доступу, перш ніж вони можуть поставити під загрозу пристрої чи мережі.
- Програмне забезпечення Cisco Adaptive Security Appliance (ASA)

Проксі-сервер

Проксі-сервер, вбудований у програмне забезпечення брандмауера, діє як безпечний шлюз між мережею та Інтернетом. Перевіряючи та контролюючи трафік на рівні програми, він блокує зовнішні загрози, приховує внутрішні IP-адреси, забезпечує дотримання політик доступу та пришвидшує продуктивність завдяки кешуванню, забезпечуючи глибоку перевірку та захист від шкідливого програмного забезпечення та несанкціонованого доступу.

Переваги:

- Захищає внутрішню мережу, маскуючи IP-адреси, запобігаючи прямому впливу зовнішніх загроз.
- Забезпечує дотримання політик, блокуючи доступ до обмежених веб-сайтів, контенту або користувачів (наприклад, соціальних мереж).
- Покращує продуктивність, зберігаючи часто використовуваний контент локально, зменшуючи використання пропускну здатності та пришвидшуючи доставку.

Stateful Inspection (SPI)

Брандмауери Stateful Inspection (SPI) підтримують динамічну таблицю станів, яка записує деталі з'єднання, включаючи IP-адреси, порти та стан сеансу. Аналізуючи контекст трафіку, SPI дозволяє дійсний зворотний трафік (наприклад, веб-відповіді), відхиляючи небажані або аномальні пакети. Така контекстна обізнаність забезпечує кращий рівень безпеки порівняно з фільтрацією без збереження стану, ефективно зменшуючи такі загрози, як підміна IP-адрес та спроби несанкціонованого доступу.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Переваги:

- Перевірка стану виявляє аномалії шляхом перевірки трафіку на відповідність записам активних сеансів, запобігаючи підробленим відповідям.
- Відстежує стан зв'язку, включаючи TCP-підтвердження та перебіг сеансу, для забезпечення дотримання точних правил безпеки.
- Підтримує розширені протоколи, такі як FTP, які покладаються на кілька одночасних підключень, забезпечуючи належну перевірку та контроль.

Консоль керування брандмауером

Консоль керування брандмауером надає централізований інтерфейс для налаштування, моніторингу та обслуговування кількох брандмауерів. Вона дозволяє адміністраторам застосовувати політики безпеки, правила аудиту та переглядати журнали трафіку та загроз, покращуючи відповідність вимогам та ефективність у складних мережах. Завдяки інформаційним панелям та функціям звітності FMC спрощує нагляд та зміцнює безпеку організації.

Переваги:

- Встановлюйте та застосовуйте правила, які визначають, який трафік, порти чи користувачі дозволені або заблоковані.
- Спостерігайте за мережевою активністю в режимі реального часу, виявляйте шкідливі програми або вторгнення та збирайте журнали для перегляду.
- Перевіряйте схеми та правила трафіку, щоб брандмауер працював безперебійно та без уповільнень.

Контроль на основі політик

Керування на основі політик трансформує управління брандмауерами, замінюючи статичні правила динамічними, централізовано керованими політиками. Це дозволяє організаціям забезпечувати безпеку на детальному рівні на основі користувачів, програм і контенту, одночасно масштабуючи захист у складних мережах. Результатом є розумніша, орієнтована на бізнес безпека, яка знижує ризики та спрощує дотримання вимог.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Переваги:

- Перевірка на основі політик дозволяє детально контролювати програми, дозволяючи або обмежуючи функції незалежно від використання портів.
- Правила можуть включати динамічні умови, такі як геолокація, час доби, класифікація пристроїв та стан відповідності кінцевих точок.
- Політики виходять за рамки фільтрації трафіку та застосовують багаторівневий захист, включаючи запобігання вторгненням, виявлення шкідливого програмного забезпечення, категоризацію URL-адрес та запобігання втраті даних.

Висока доступність

Висока доступність (НА) у програмному забезпеченні брандмауерів гарантує безперебійну безпеку мережі шляхом розгортання резервних брандмауерів у кластерних системах. Вони працюють в основному/резервному режимі або в режимі розподілу навантаження, використовуючи сигнали серцевого ритму та синхронізовані конфігурації для забезпечення безперебійного перемикання на резервний ПК, мінімізації простоїв та підтримки безперервності бізнесу.

Переваги:

- Усуває перебої в мережі, забезпечуючи безперебійний доступ для операцій, що потребують постійного підключення.
- Зберігає стабільне забезпечення безпеки навіть під час системних збоїв, зменшуючи ризики.
- Активні розгортання розподіляють трафік між кількома брандмауерами, підвищуючи пропускну здатність та швидкість реагування.

Вибір найкращих функцій для програмного забезпечення брандмауера

Програмне забезпечення брандмауера функціонує як захисний шар для мереж і пристроїв, постійно перевіряючи вхідний і вихідний трафік. Він забезпечує дотримання політик безпеки, фільтруючи пакети даних, запобігаючи

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

вторгненням і зменшуючи загрози. Це забезпечує безпечну роботу програм, незалежно від того, розгорнуті вони на персональних комп'ютерах, корпоративних серверах чи хмарних платформах. Важливо вибрати відповідне програмне забезпечення в робочому просторі, яке пропонує правильні функції, необхідні для цього робочого простору.

3.2 Розробка структурної схеми

Традиційна фільтрація пакетів, можливо, була достатньою у 80-х та 90-х роках, але не в сучасному світі. Коли йдеться про типи кіберзагроз, з якими зараз стикається бізнес, програмне забезпечення для глибокої перевірки пакетів (DPI) та перевірка стану пакетів (SPI) є гарячими темами. Як і слід було очікувати, вони мають деякі унікальні відмінності. Розуміння їхніх ролей є вирішальною частиною вибору правильних брандмауерів, аналізу бізнес-потреб та навігації архітектурою безпеки вашої мережі. Якщо ви хочете уникнути відчуття перевантаженості через огляд та аналіз мережі, вам слід врахувати важливість DPI та SPI. У цьому розділі порівнюються ці два параметри, щоб отримати чітку відповідь на питання, чим вони відрізняються та чому обидва важливі в сучасному ландшафті мережевої безпеки.

Важлива роль перевірки пакетів

Вам потрібна видимість мережі, яка виходить за рамки простої фільтрації портів та IP-адрес. Інспекція пакетів зосереджується на виявленні набагато складніших загроз. Це просто поточний стан потенційних загроз безпеці мережі, тому DPI та SPI так поширені в наш час. Вони також є важливим компонентом детальних політик безпеки та підтримують набагато більший контроль над мережевим трафіком загалом. З часом це стає лише важливішим, оскільки відіграє певну роль у багатьох вимогах мережевої безпеки, таких як усунення несправностей, забезпечення відповідності вимогам та видимість у більш динамічних мережевих конфігураціях.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Що таке глибока перевірка пакетів?

З огляду на те, наскільки технічно складними можуть бути засоби безпеки, деякі характеристики та функції легко пропустити. DPI працює з повним вмістом пакетів і працює на рівні програми для набагато глибшого огляду. Поверхневий огляд більше не є достатнім, що робить DPI необхідністю для безпеки вашої мережі.

Ось ще кілька деталей про DPI, які вам слід знати:

- Перевірка пакетів проходить кілька етапів, таких як перевірка заголовків, декодування корисного навантаження, порівняння сигнатур загроз тощо.
- Методи, специфічні для DPI, включають аналіз поведінки, зіставлення зі зразками, перевірку протоколів та виявлення статистичних аномалій.
- Виявляє шкідливий контент, який може бути прихований серед звичайного мережевого трафіку.

Хоча цього може здатися більш ніж достатнім, поряд з вашими апаратними та програмними рішеннями, не можна забувати про роль SPI тут.

Що таке перевірка стану пакетів?

Основна відмінність SPI полягає в тому, що він зосереджується на контексті мережевого трафіку. Це означає, що він гарантує, що кожен пакет є частиною дійсного, існуючого сеансу. Це набагато краще, ніж те, що може забезпечити фільтрація без урахування стану, і добре захищає вашу мережу від атак, пов'язаних із з'єднанням.

Нижче наведено кілька ключових характеристик, унікальних для перевірки пакетів з урахуванням стану:

- Захищає від певних атак, таких як захоплення сеансу та підміна TCP
- Класифікує трафік за різними станами для виявлення потенційно шкідливої активності
- Використовує «таблицю станів» для реєстрації важливих деталей мережевого трафіку, таких як номери портів, IP-адреси, час сеансу тощо.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Хоча це правда, що SPI дещо менш ретельний, ніж такі методи, як DPI, він відомий своєю ефективністю в таких діях, як управління безпекою на рівні з'єднання. Без цього розподілу між цими двома функціями новим і зростаючим компаніям може бути важко зрозуміти їхню важливість.

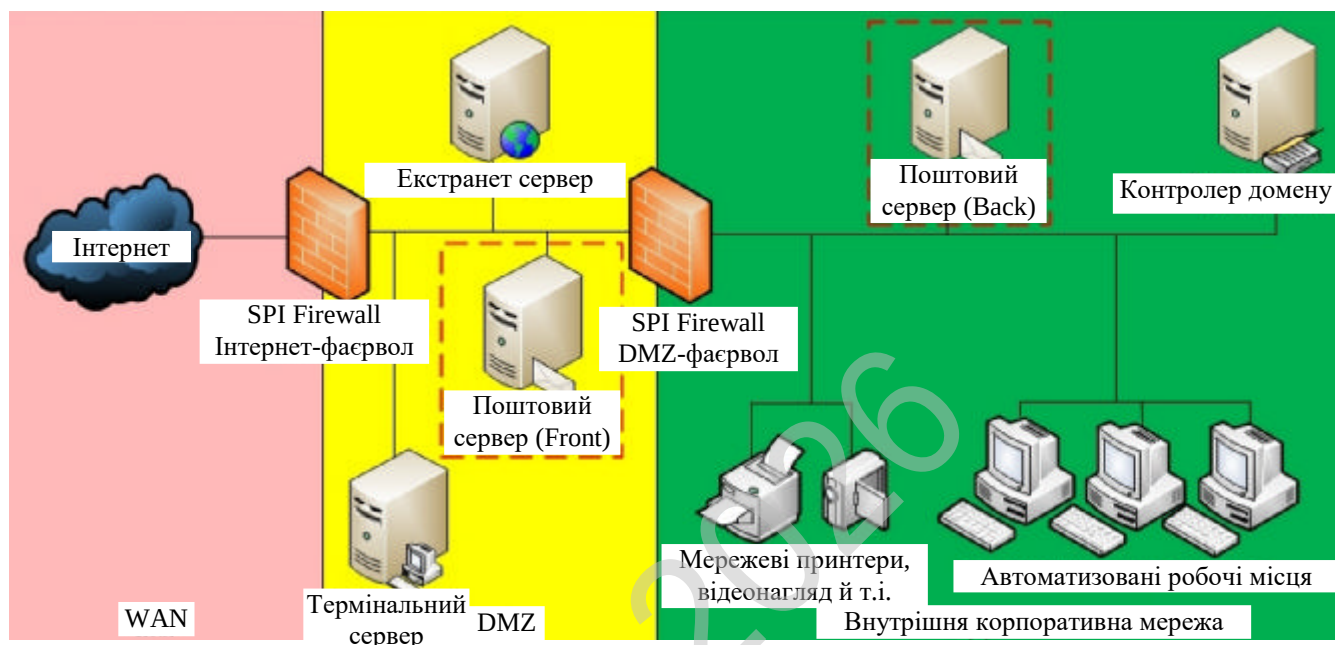


Рисунок 3.1 – Структурна схема системи

Прозорість у DPI проти SPI

З огляду на функції, можливості та загальну інтеграцію у вашу мережеву безпеку, порівняння DPI та SPI може бути досить заплутаним. Це не означає, що деталі не мають значення, але безумовно існує простіший спосіб засвоїти інформацію.

Річ не в тому, який з них обрати, а в тому, чому DPI та SPI важливі в сучасному ландшафті кібербезпеки. На щастя, багато з цього охоплено апаратним забезпеченням мережевої безпеки, ліцензуванням і навіть керованими послугами.

Давайте підсумуємо

Існує багато технічного жаргону, пов'язаного з брандмауерами та мережевою безпекою, але ігнорування цього може зробити ваш бізнес повністю

вразливим до атак. Якщо врахувати масштаби кіберзагроз, які існують, найкраще бути на крок попереду.

Для багатьох молодих і зростаючих компаній це легше сказати, ніж зробити, оскільки вони проходять необхідний етап навчання. Ніколи не завадить отримати невелику допомогу на цьому шляху, і саме тут можуть допомогти члени нашої команди.

3.3 Розробка функціональної схеми

Брандмауер бездротового маршрутизатора перевіряє пакет на рівні мережі, зберігає відомості про сеанси TCP і UDP, включаючи тайм-аути й кількість активних сеансів, а також надає можливість визначати й запобігати певні типи мережних атак.

На сторінці «Виявлення вторгнення» є наступні розділи:

- «Функція виявлення вторгнення».
- «Контроль пакетів з веденням статистики».
- «Попередження електронною поштою бездротовим маршрутизатором, при спробі несанкціонованого входу в мережу».
- «Політика з'єднання».
- «Критерії визначення атак з метою відключення».

Виберіть параметри виявлення вторгнення й не забудьте натиснути кнопку «Зберегти налаштування» по закінченню.

Функція виявлення вторгнення

Використовуйте цей розділ сторінки «Виявлення вторгнення», щоб включити наступні типи захисту:

- «Статистика «SPI» і захист брандмауера проти атак з метою відключення»: обмежує доступ для вхідного трафіку на порту WAN. Якщо функція SPI включена, то всі вхідні пакети будуть блокуватися, крім тих типів, які обрані в розділі «Контроль пакетів з веденням статистики».

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– «Несправність RIP»: запобігає переповнення вхідних черг через скупчення пакетів. Якщо пакет запиту RIP не підтверджений маршрутизатором, він звичайно залишається у вхідній черзі й не буде звільнений. Пакети, що накоплюються, можуть викликати переповнення вхідної черги, що приведе до серйозних проблем для всіх протоколів. Включення параметра «Несправність RIP» запобігає скупчування пакетів.

– «Скасувати запити Ping на WAN»: скасування відправлення запитів ping з порту WAN бездротового маршрутизатора в мережу.

Контроль пакетів з веденням статистики

Мережний екран Stateful Packet Inspection «SPI» дозволяє захистити мережу від небажаного трафіку шляхом виконання наступних завдань:

– Перевірка всіх пакетів даних для визначення, чи запитував установлений комп'ютер з'єднання, перед тим, як пропустити пакети через брандмауера.

– Закриття портів, поки не буде запитане з'єднання з певним портом.

Щоб включити SPI:

У розділі «Функція виявлення вторгнення» виберіть «Статистика «SPI» і захист брандмауера проти атак з метою відключення».

Виберіть тип або типи трафіку, припустимі для брандмауера: «Фрагментація пакетів», «З'єднання TCP», «Сеанс UDP», «Служба FTP» або «Служба TFTP».

Припустимими будуть уважатися тільки обрані трафіки, ініційовані із внутрішньої локальної мережі. Наприклад, якщо обрано тільки параметр «Служба FTP» у розділі Stateful Packet Inspection «Контроль пакетів з веденням статистики», те весь вхідний трафік буде блокуватися, крім FTP-з'єднань, ініційованих з локальної мережі.

Натисніть кнопку «Зберегти налаштування».

Попередження електронною поштою бездротовим маршрутизатором, при спробі несанкціонованого входу в мережу

Використовуйте цей розділ для налаштування адреси електронної пошти, за якій потрібно одержувати повідомлення про спроби несанкціонованого доступу.

Заповніть всі поля, що ставляться до конфігурації електронної пошти:

– «Адреса електронної пошти»: адреса електронної пошти, на який повинні надходити повідомлення від брандмауера.

– «Адреса сервера SMTP»: адреса сервера SMTP.

– «Адреса сервера POP3»: уведіть адресу використовуваного сервера POP3. Потрібно тільки в тому випадку, якщо для відправлення повідомлень електронною поштою поштовий сервер повинен перевірити дійсність даних.

– «Ім'я користувача»: ім'я користувача для сервера POP3. Потрібно тільки в тому випадку, якщо для відправлення повідомлень електронною поштою поштовий сервер повинен перевірити дійсність даних.

– «Пароль»: пароль, необхідний для сервера POP3. Потрібно тільки в тому випадку, якщо для відправлення повідомлень електронною поштою поштовий сервер повинен перевірити дійсність даних.

Натисніть кнопку «Зберегти налаштування».

Політика з'єднання

Уведіть відповідні значення для сеансів TCP/UDP:

– «Очікування напіввідчиленої фрагментації»: число секунд, протягом яких структура стану пакета залишається активною. При витіканні тайм-ауту маршрутизатор скидає незібраний пакет, звільняючи структуру для іншого пакета.

– «Очікування TCP SYN»: визначення часу очікування програмним забезпеченням сеансу TCP для синхронізації перед скиданням сеансу.

– «Очікування TCP FIN»: визначення часу підтримки сеансу TCP після виявлення брандмауером пакета FIN.

– «Тайм-аут простою з'єднання TCP»: період часу, протягом якого буде здійснюватися керування сеансом TCP при відсутності активності.

– «Тайм-аут простою сеансу UDP»: період часу, протягом якого буде здійснюватися керування сеансом UDP при відсутності активності.

– «Тайм-аут простою каналу даних H.323»: період часу, протягом якого буде здійснюватися керування сеансом H.323 при відсутності активності.

Натисніть кнопку «Зберегти налаштування».

Критерії визначення атак з метою відключення

Мережні атаки, які скасовують доступ до мережного пристрою, називаються атаками з метою відключення «DoS». Атаки з метою відключення спрямовані на пристрої й мережі, підключені до Інтернету. Їхньою метою є не крадіжка даних, а відключення пристрою або мережі, щоб користувачі не мали доступ до мережних ресурсів. Бездротовий маршрутизатор забезпечує захист від наступних атак з метою відключення:

- IP-спуфинг.
- Атаки Land.
- Смертельний пінг.
- IP нульової довжини.
- Смурфинг.
- Закольцовування порту UDP.
- Атаки Snork.
- Нульове сканування TCP.
- Лавинна маршрутизація TCP з'єднань.

Уведіть відповідні значення для критеріїв виявлення "відмови в обслуговуванні" «DoS» і сканування портів:

– «Найбільше загальне число незавершених сеансів TCP/UDP»: визначення числа знову встановлених сеансів, по досягненні якого програма почне видалення напіввідчинених сеансів.

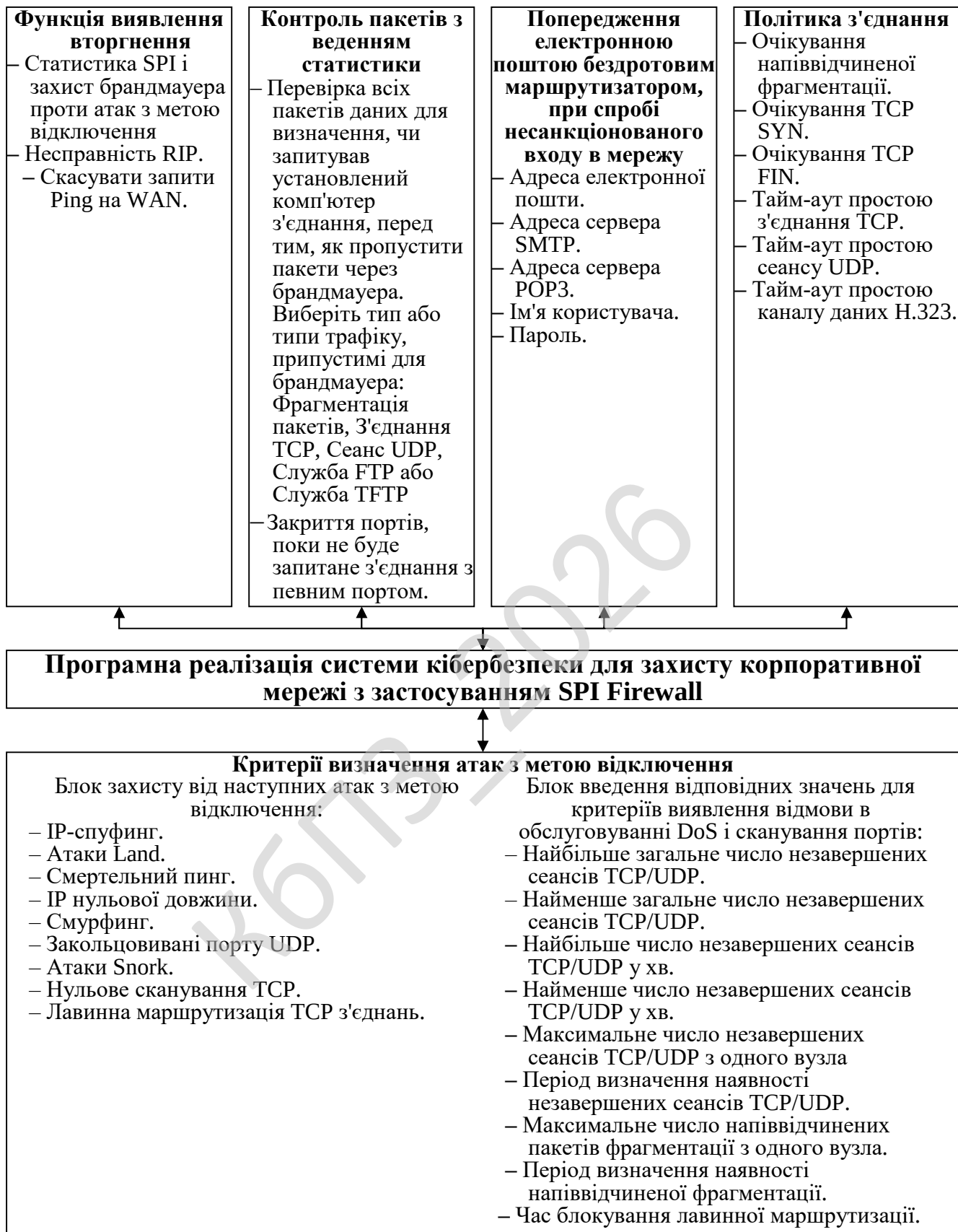


Рисунок 3.2 Функціональна схема системи

– «Найменше загальне число незавершених сеансів TCP/UDP»: визначення числа знову встановлених сеансів, по досягненні якого програма припинить видалення напіввідчинених сеансів.

– «Найбільше число незавершених сеансів TCP/UDP «у хв»»: максимальне число припустимих незавершених сеансів TCP/UDP у хвилину.

– «Найменше число незавершених сеансів TCP/UDP «у хв»»: мінімальне число припустимих незавершених сеансів TCP/UDP у хвилину.

– «Максимальне число незавершених сеансів TCP/UDP з одного вузла»: максимальне число незавершених сеансів TCP/UDP з одного вузла.

– «Період визначення наявності незавершених сеансів TCP/UDP»: період часу, протягом якого незавершений сеанс TCP/UDP визначається як незавершений.

– «Максимальне число напіввідчинених пакетів фрагментації з одного вузла»: максимальне число напіввідчинених пакетів фрагментації з одного вузла.

– «Період визначення наявності напіввідчиненої фрагментації»: період часу, протягом якого сеанс напіввідчиненої фрагментації визначається як напіввідчинений.

– «Час блокування лавинної маршрутизації»: період часу від виявлення лавинної маршрутизації до блокування атаки.

Натисніть кнопку «Зберегти налаштування».

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

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



Рисунок 3.3 – Діаграма взаємодії процесів

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-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

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-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

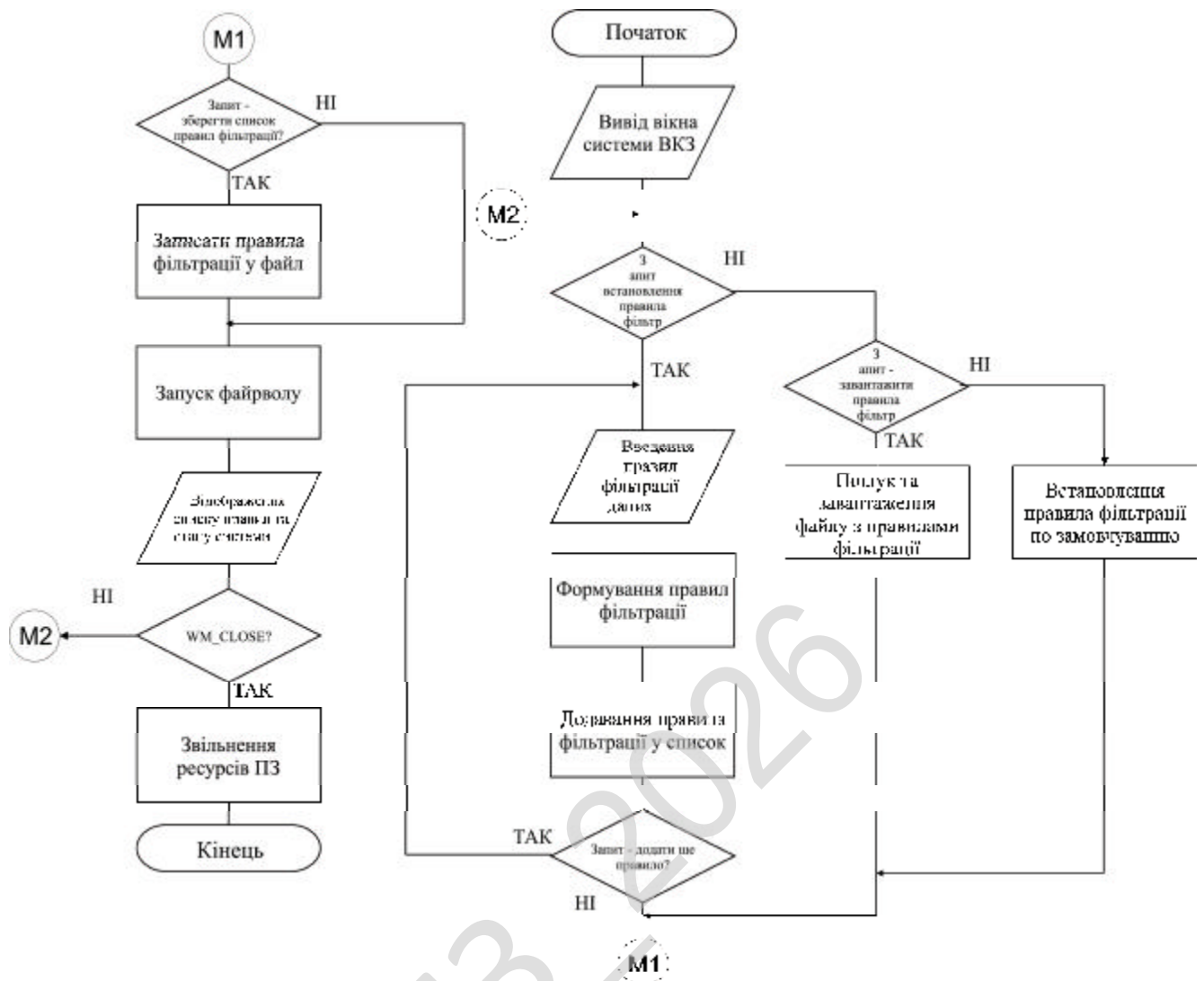


Рисунок 4.1 – Блок-схема основної програми

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані.

API є абстрактним поняттям — програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

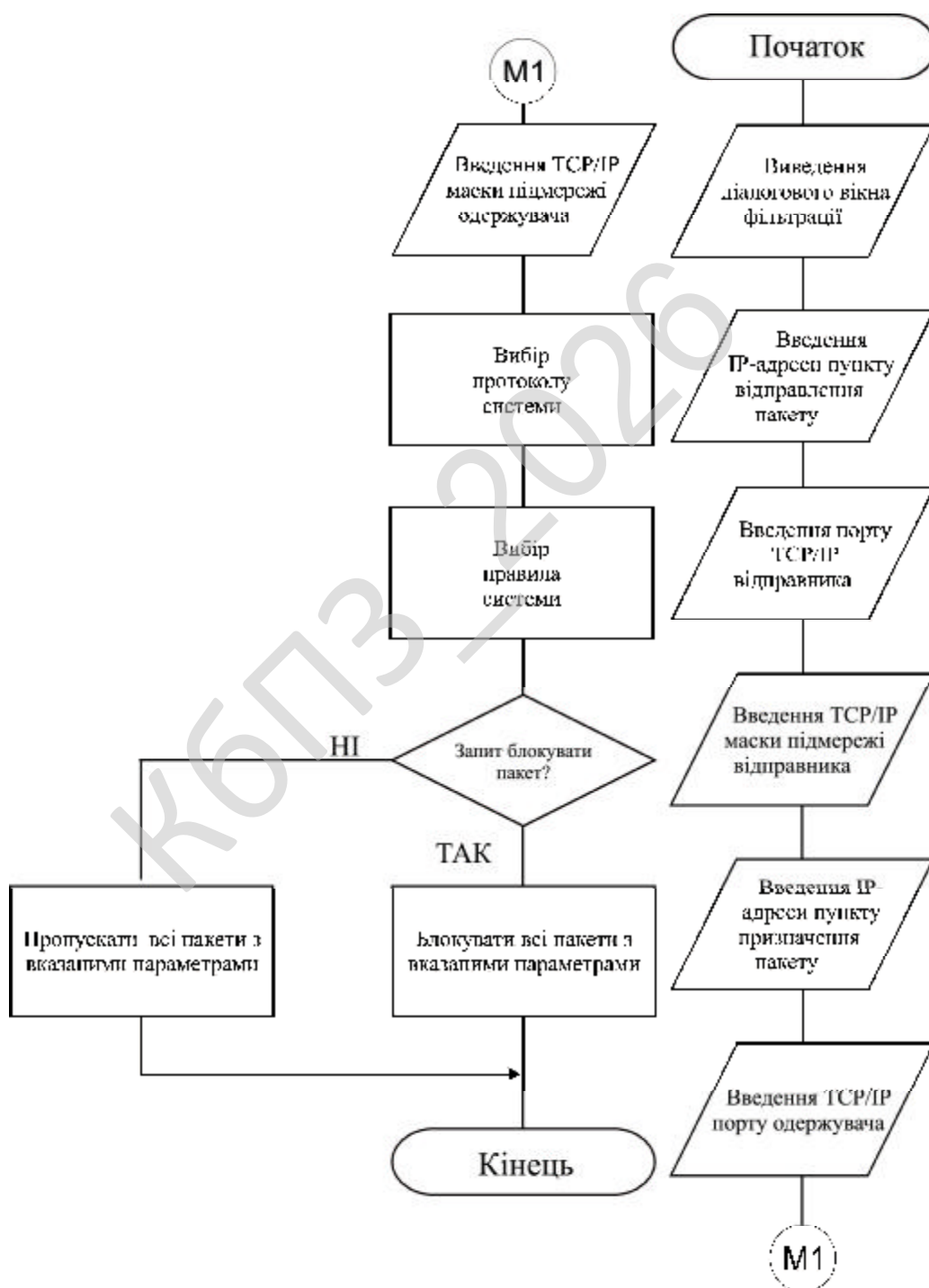


Рисунок 4.2 – Блок-схема роботи підпрограми

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППП, як правило, пов'язаний із бібліотеками програмного забезпечення: ППП описує і вказує очікувану поведінку в той час, як бібліотека є фактичною реалізацією даного набору правил.

Один ППП може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППП, але на відміну від звичайного використання ППП, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППП реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППП може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППП, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППП, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

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

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного

сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів.

Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

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

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

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

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

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація.

У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

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

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

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

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер.

Це позбавляє його необхідності завантажувати та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

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

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

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

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються.

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

Розширювана мова розмітки (Extensible Markup Language, скорочено XML) – запропонований консорціумом World Wide Web Consortium (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. Є спрощеною підмножиною мови розмітки SGML.

XML-документ складається із текстових знаків, і придатний до читання людиною. Стандарт XML (Recommendation, перше видання від 10 лютого 1998, останнє, четверте видання 29 вересня 2006) визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів.

Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних. Ці обмеження описуються мовами схем (Schema), такими як XML Schema (XSD), DTD або RELAX NG. Прикладами мов, заснованих на XML, є: XSLT, XAML, XUL, RSS, MathML, GraphML, XHTML, SVG, а також XML Schema.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Основні поняття. Коректність.

Коректний документ (well-formed document) відповідає всім синтаксичним правилам XML. Документ, що не є коректним, не може називатись XML-документом. Сумісний синтаксичний аналізатор (Conforming parser) не повинен обробляти такі документи. Зокрема, коректний XML-документ має:

1. Лише один елемент у корені.

2. Непорожні елементи розмічено початковим та кінцевим тегами (наприклад, <пункт> Пункт 1</пункт>). Порожні елементи можуть позначатися «закритим» тегом, наприклад <IAmEmpty />. Така пара еквівалентна <IAmEmpty></IAmEmpty>.

3. Один елемент не може мати декілька атрибутів з однаковою назвою. Значення атрибутів перебувають або в одинарних ('), або у подвійних (") лапках.

4. Теги можуть бути вкладені, але не можуть перекриватись. Кожен некореневий елемент мусить повністю перебувати в іншому елементі.

5. Документ має складатися тільки з правильно закодованих дозволених символів Юнікоду. Єдиними кодуваннями, які обов'язково має розуміти XML-процесор, є UTF-16 та UTF-8. Фактичне та задеклароване кодування (character encoding) документа мають збігатись. Кодування може бути задекларовано ззовні, як у заголовку «Content-Type» при передачі по протоколу HTTP, або в самому документі використанням явної розмітки на самому початку документа. У разі відсутності інформації про кодування, документ має бути в кодуванні UTF-8 (або його підмножині ASCII).

Валідність

Документ називається валідним (valid), якщо він є коректним, містить посилання на граматичні правила та повністю відповідає обмеженням, вказаним у цих правилах (DTD або XML Schema або іншому подібному документі).

Синтаксичний аналізатор

Синтаксичним аналізатором (часто парсер, від parser) називається програма або компонент, що читає XML-документ, проводить синтаксичний

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

аналіз та відтворює його структуру. Якщо синтаксичний аналізатор перевіряє документ на валідність, то такий аналізатор називають валідатором (validating).

Назви елементів чутливі до регістру літер. Наприклад, наступна пара елементів правильна: `<Step> ... </Step>` у той час як ця – ні: `<Step> ... </step>`.

Правильний вибір назв для XML-елементів підкреслюватиме значення даних у створеній мові розмітки. Це сприятиме полегшенню роботи людей з такими документами, зберігаючи можливості для комп'ютерної обробки даних.

Вибір змістовних назв передає семантику елементів та атрибутів для людини, без посилання на зовнішню документацію. Однак це може призвести до надмірності розмітки, що ускладнює редагування й збільшує розмір файлів.

Правильний вибір назв для XML-елементів підкреслюватиме значення даних у створеній мові розмітки. Це сприятиме полегшенню роботи людей з такими документами, зберігаючи можливості для комп'ютерної обробки даних. Вибір змістовних назв передає семантику елементів та атрибутів для людини, без посилання на зовнішню документацію. Однак це може призвести до надмірності розмітки, що ускладнює редагування й збільшує розмір файлів. XML-документи мають як фізичну, так і логічну структуру.

Приклад XML-документа:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.3/" xml:lang="uk">
  <page>
    <title>Фукідід</title>
    <id>1529</id>
    <revision>
      <id>4382</id>
      <timestamp>2006-09-18T22:11:53Z</timestamp>
      <minor />
      <comment>Interwiki</comment>
      <text xml:space="preserve">{{Wikipedia}}
    </text>
    </revision>
  </page>
</mediawiki>
```

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Фізична структура

Сутності (Entity). Головною сутністю є зміст документа. Інші можливі сутності вказуються за допомогою. Посилання на сутності (&назва; в самому документі, та, наприклад %назва; у визначенні його типу) можуть слугувати в ролі позначень спеціальних символів, посилань на спеціальні символи або окремих документів чи фрагментів тексту.

XML-декларація, в ній вказується версія XML, кодування та інша допоміжна інформація.

Декларація типу документа може застосовуватись для того, аби додавати нові типи сутностей та визначати логічну структуру документа.

Логічна структура

XML-документ має ієрархічну логічну структуру, і може представлятись у вигляді дерева. Вузлами цього дерева можуть бути: елементи, фізична структура яких складається із коректної пари відкриваючого та закриваючого тегів (<Назва-тега>) та (</Назва-тега>), або тега порожнього елемента (<Назва-тега />).

Атрибути, що мають вигляд пар ключ-значення (назва атрибута="значення атрибута") і знаходяться або у відкриваючому, або у порожньому тезі (подібно до метаданих).

Вказівки щодо обробки документа (Processing Instruction) (<?Обробник параметр ?>).

Коментарі (<!-- Текст коментаря -->).

Текст, або у вигляді простого тексту, або фрагментів CDATA (<![CDATA[довільний текст]]>).

XML-документ повинен мати лише один кореневий елемент. Решта елементів є піделементами цього кореневого елемента. Деякі веб-браузери здатні безпосередньо відображати XML-документи. Це може досягатись шляхом застосування таблиці стилів (Stylesheet). Вказані у таблиці стилів операції можуть призводити до перетворення XML-документа в інший, відмінний від XML формат.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою національного стандарту захисту інформації на основі алгоритму шифрування/дешифрування ДСТУ 4145-2002 з використанням еліптичних кривих над двійковим розширеним полем Галуа. У системі шифрування/дешифрування як параметри розглядається еліптична крива $E_p(a,b)$ і точка G на ній. Учасник В вибирає закритий ключ n і обчислює відкритий ключ $P_B = n \times G$. Щоб зашифрувати повідомлення P_m використовується відкритий ключ одержувача В P_B . Учасник А вибирає випадкове ціле позитивне число k і обчислює зашифроване повідомлення C_m , що є точкою на еліптичній кривій.

$$C_m = \{k \times G, P_m + k \times P_B\}. \quad (4.1)$$

Щоб дешифрувати повідомлення, учасник В множить першу координату точки на свій закритий ключ і віднімає результат від другої координати:

$$P_m + k \times P_B - n_B \times (k \times G) = P_m + k \times (n_B \times G) - n_B \times (k \times G) = P_m. \quad (4.2)$$

Учасник А зашифрував повідомлення P_m додаванням до нього $k \times P_B$. Ніхто не знає значення k , тому, хоча P_B і є відкритим ключем, ніхто не знає $k \times P_B$. Супротивнику для відновлення повідомлення доведеться обчислити k . Зробити це буде нелегко. Одержувач також не знає k , але йому як підказку посилається $k \times G$. Помноживши $k \times G$ на свій закритий ключ, одержувач одержить значення, що було додано відправником до незашифрованого повідомлення. Тим самим одержувач, не знаючи k , але маючи свій закритий ключ, може відновити незашифроване повідомлення.

Нехай задано просте число $p > 4$. Тоді еліптичною кривою E , визначеною над розширеним двійковим полем F_{2^m} , називається безліч пар чисел (x, y) , $x, y \in F$, що задовольняють тотожності:

$$y^2 \equiv x^3 + a \cdot x + b \pmod{2^m}, \quad (4.3)$$

де $4 \cdot a^3 + 27 \cdot b^2$ не рівно з нулю по модулю 2^m .

Інваріантом еліптичної кривої називається величина $J(E)$, що задовольняє

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

тотожності:

$$J(E) \equiv 1728 \frac{4a^3}{4a^3 + 27b^2} \pmod{2^m}. \quad (4.4)$$

Коефіцієнти a , b еліптичної кривої E , по відомому інваріанту $J(E)$, визначаються таким чином:

$$\begin{cases} a \equiv 3k \pmod{2^m} \\ b \equiv 2k \pmod{2^m} \end{cases} \text{ де } k \equiv \frac{J(E)}{1728 - J(E)} \pmod{2^m}, J(E) \neq 0 \text{ або } 1728. \quad (4.5)$$

Пари (x, y) , що задовольняють тотожності (4.1), називаються точками еліптичної кривої E , x та y – відповідно x - та y -координатами точки.

Точки еліптичної кривої позначатимемо $Q(x, y)$ або просто Q . Дві точки еліптичної кривої рівні, якщо рівні їх відповідні x - і y -координати.

На безлічі всіх точок еліптичною кривою E введемо операцію додавання, яку позначатимемо знаком "+". Для двох довільних точок $Q_1(x_1, y_1)$ та $Q_2(x_2, y_2)$ еліптичної кривої E , розглянемо декілька варіантів.

Нехай координати точок Q_1 та Q_2 задовольняють умові $x_1 \neq x_2$. В цьому випадку їх сумою називатимемо точку $Q_3(x_3, y_3)$ координати якої визначаються порівняннями:

$$\begin{cases} x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{2^m}, \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{2^m}, \end{cases} \text{ де } \lambda \equiv \frac{y_2 - y_1}{x_2 - x_1} \pmod{2^m}. \quad (4.6)$$

Якщо виконана рівність $x_1 = x_2$ та $y_1 = y_2 \neq 0$, то визначимо координати точки Q_3 таким чином:

$$\begin{cases} x_3 \equiv \lambda^2 - 2x_1 \pmod{2^m}, \\ y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{2^m}, \end{cases} \text{ де } \lambda \equiv \frac{3x_1^2 + a}{2y_1} \pmod{2^m}. \quad (4.7)$$

У разі, коли виконана умова $x_1 = x_2$ та $y_1 = -y_2 \pmod{p}$, суму точок Q_1 та Q_2 називатимемо нульовою точкою O , не визначаючи її x - і y -координати. В цьому випадку, точка Q_2 називається запереченням точки Q_1 . Для нульової точки O виконана рівність:

$$Q + O = O + Q = Q, \quad (4.8)$$

де Q – довільна точка еліптичної кривої E .

Щодо введеної операції складання безліч всіх точок еліптичною кривою E , разом з нульовою точкою, утворюють кінцеву абельову (комутативну) групу порядку t , для якого виконана нерівність:

$$p+1-2\sqrt{p} \leq t \leq p+1+2\sqrt{p} \quad (4.9)$$

Точка Q називається точкою кратності k , або просто – кратною точкою еліптичної кривої E , якщо для деякої точки P виконана рівність:

$$Q = P + \dots + P = kP \quad (4.10)$$

К6ПЗ-2026

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображена програмна реалізація системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Верхнього меню: Файл; Правила; Довідка.
- Розділу виведення результату роботи системи.

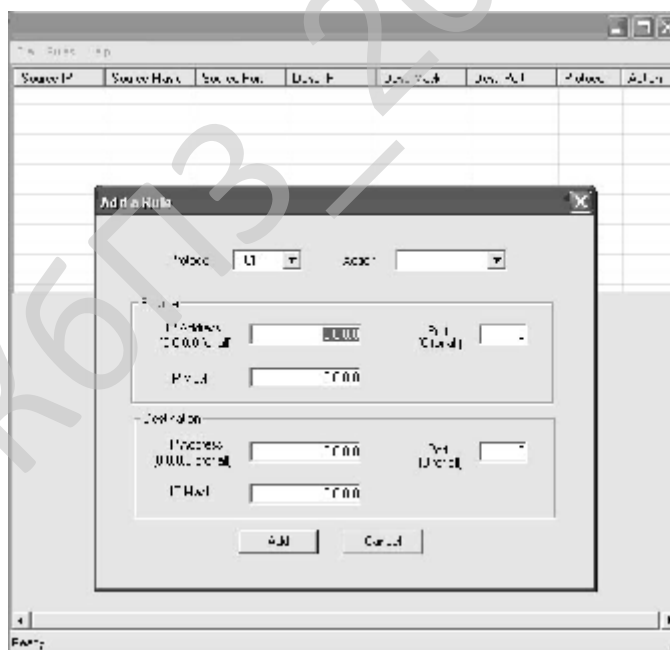


Рисунок 5.1 – Головне вікно ПЗ

Міжмережевий або мережний екран являє собою комплекс устаткування й програмних засобів, яким контролюються й фільтруються минаючі мережні пакети, з огляду на попередньо встановлені критерії.

Цей захисний екран, називаний брандмауером або Firewall, захищає від атаки хакерів локальні корпоративні мережі, запобігає несанкціонованому зовнішньому доступу, спробам зламати сховища даних і зробити крадіжку інформації.

Корпоративні користувачі, завдяки брандмауерові, одержують безпечний доступ до світової павутини, контрольоване віддалене підключення зовнішніх пристроїв до локального ІТ-інфраструктурі.

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

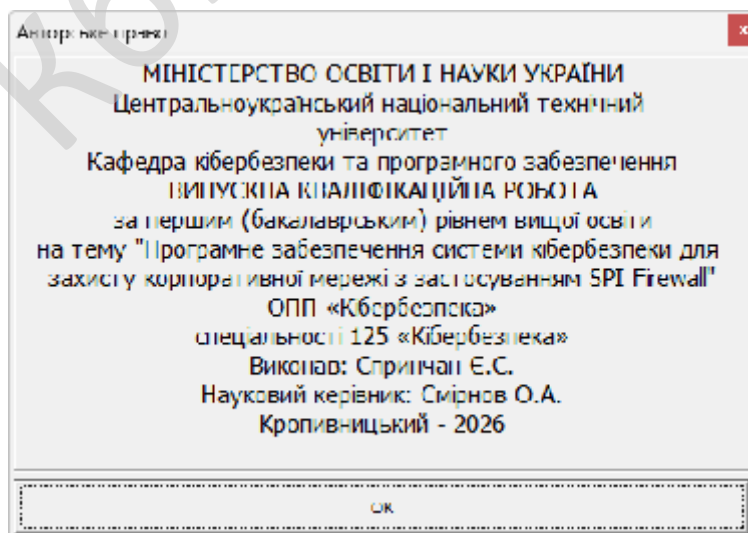


Рисунок 5.2 – Авторське право

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

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

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

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

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

Проводилась оцінка:

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

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

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

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.
- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).
- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

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

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

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

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

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», чию 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 – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

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 корпоративної мережі з застосуванням SPI Firewall.
- Досліджена система для захисту корпоративної мережі з застосуванням SPI Firewall.
- На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI Firewall.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для захисту корпоративної мережі з застосуванням SPI Firewall.

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 Visual C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для захисту корпоративної мережі з застосуванням SPI

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Firewall. 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 ДСТУ 4145-2002.

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-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

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., 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.
14. 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

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

15. 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

16. 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.

17. 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.

18. 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.

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

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

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

29. 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

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

31. 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.

32. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: *Rajakumar, G., Du, K.L., 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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,

39. 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.

40. 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.

41. 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.

					ВКРБ-125.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

42. 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.

43. 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.

44. 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

45. 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.

46. 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.

47. 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.

48. 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.

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

50. 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.

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

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

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

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

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

56. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-

quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

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

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

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