Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Купіс Валентин Андрійович

на здобуття ступеня вищої освіти Бакалавра

Система виявлення та протидії загрозам для USB-інтерфейсів

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

-

КРБКБ.220111.22.01.08 ПЗ

Виконав студент 4 курсу, група КБ-22-1 _____ Валентин КУПІС
Ініціали, прізвище

Керівник канд. тех. наук, доцент _____ Володимир ДЖУЛІЙ
Науковий ступінь, вчене звання — Ініціали, прізвище

Нормоконтролер д-р філософії _____ Наталія ПЕТЛЯК
Науковий ступінь, вчене звання — Ініціали, прізвище

До захисту допускаю:

Зав. кафедри кібербезпеки _____ Юрій КЛЬОЦ
Ініціали, прізвище

17 06 2026р.

Хмельницький, 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

9 січня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Купіс Валентин Андрійович

1 Тема роботи Система виявлення та протидії загрозам для USB-інтерфейсів

Керівник роботи канд.техн.наук, доцент Джулій Володимир Миколайович

Затверджено наказом ректора університету від 8 січня 2026 № 7

2 Строк подання студентом кваліфікаційної роботи на кафедру _____

3 Вихідні дані до роботи розробити систему виявлення та протидії загрозам для USB-інтерфейсів; дослідити типові атаки через USB-пристрої; проаналізувати наявні засоби контролю знімних носіїв; спроєктувати апаратно-програмний комплекс з білим списком, журналюванням та двома режимами роботи; виконати тестування працездатності й ефективності системи.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) вступ; аналіз загроз інформаційній безпеці через USB-інтерфейси; проєктування апаратно-програмного комплексу; розробка, розгортання та тестування системи; висновки; перелік джерел; додатки.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)Структура автокодувальника LSTM. Попередня обробка наборів даних. Матриці плутанини.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 12 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Травень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Травень	
Захист КР	Червень	

Студент _____ Валентин КУПІС

Керівник кваліфікаційної роботи _____ Володимир ДЖУЛІЙ

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система виявлення та протидії загрозам для USB-інтерфейсів.

Автор роботи: Купіс Валентин Андрійович.

Керівник роботи: Джулій Володимир Миколайович.

Пояснювальна записка: 66 сторінок, 1 додаток, 6 рисунків, 6 таблиць 41 джерел.

Графічна частина: 5 плакатів.

Ключові слова: USB-інтерфейс, USB-пристрій, HID-ін'єкція, апаратно-програмний комплекс, USB-фільтр, довірені пристрої, виявлення загроз

У сучасних умовах активного використання USB-пристроїв у комп'ютерних системах важливим завданням є забезпечення контролю їх підключення та запобігання загрозам, що можуть реалізовуватися через USB-інтерфейси. У кваліфікаційній роботі розроблено апаратно-програмний комплекс для виявлення та протидії загрозам, пов'язаним із використанням зовнішніх USB-пристроїв. Запропоноване рішення спрямоване на контроль підключення пристроїв, перевірку їх належності до довірених, виявлення потенційно небезпечної активності та реагування на інциденти безпеки. У роботі розглянуто основні типи атак із використанням USB-інтерфейсів, зокрема BadUSB, HID-ін'єкції, USB Drop Attack, підміну дескрипторів пристроїв, несанкціонований витік даних через знімні носії та фізичне пошкодження обладнання. Розроблений комплекс включає мікроконтролерний USB-фільтр, програмний агент моніторингу, базу довірених пристроїв, механізм застосування політик реагування та шифроване журналювання подій. Система передбачає два режими роботи: апаратно-програмний із використанням модуля контролю USB-з'єднань та програмний для робочих станцій, де встановлення апаратного фільтра є неможливим або недоцільним. Практичне значення роботи полягає у можливості використання розробленого рішення для підвищення рівня захищеності робочих місць малого та середнього бізнесу і навчальних лабораторій.

Дата: 25.05.2026

ANNOTATION

Topic of the qualification work: System for Detecting and Countering Threats to USB Interfaces.

Author of the work: Kupis Valentin Andriyovych

Mentor: Dzhuliy Volodymyr Mykolayovych

Total volume of work: 66 pages, 1 appendice, 6 figures, 6 tables, 41 links.

Graphic part: 5 posters.

Keywords: USB interface, USB device, BadUSB, hardware-software system, USB filter, trusted devices, threat detection, incident response, cybersecurity.

In the current conditions of widespread use of USB devices in computer systems, an important task is to ensure control over their connection and prevent threats that may be carried out through USB interfaces. In the qualification work, a hardware and software complex for detecting and countering threats associated with the use of external USB devices was developed. The proposed solution is aimed at controlling device connections, verifying whether connected devices belong to the category of trusted ones, detecting potentially dangerous activity, and responding to security incidents. The work considers the main types of attacks using USB interfaces, including BadUSB, HID injection, USB Drop Attack, device descriptor spoofing, unauthorized data leakage through removable storage media, and physical damage to equipment. The developed complex includes a microcontroller-based USB filter, a software monitoring agent, a database of trusted devices, a mechanism for applying response policies, and encrypted event logging. The system provides two operating modes: a hardware and software mode using a USB connection control module and a software mode for workstations where the installation of a hardware filter is impossible or impractical. The practical value of the work lies in the possibility of using the developed solution to improve the security level of workplaces in small and medium-sized businesses and educational laboratories

Дата: 25.05.2026 _____

ЗМІСТ

ВСТУП

У межах цієї роботи USB-інтерфейс розглядається не лише як фізичний порт для передавання даних, а як комплексний канал взаємодії між користувачем, операційною системою, драйверами, мікроконтролером пристрою та політиками безпеки організації. Саме така багаторівнева природа робить USB-захист складним завданням: один і той самий пристрій може одночасно виконувати функції накопичувача, клавіатури, мережевого адаптера та службового інтерфейсу керування.

Проблема ускладнюється тим, що більшість операційних систем історично орієнтовані на зручність підключення периферії. Механізм Plug and Play автоматизує розпізнавання пристрою, завантаження драйверів та надання доступу до системних ресурсів. Для звичайного користувача це є перевагою, але для захисної системи створює часовий проміжок, у якому шкідливий пристрій може встигнути виконати команди до того, як адміністратор або антивірусний засіб відреагує на інцидент.

Тому ефективна система протидії USB-загрозам повинна працювати до моменту фактичного надання пристрою доступу до операційної системи або принаймні синхронно з подією підключення. У роботі обґрунтовується поєднання апаратного контролю, який відсікає частину загроз на рівні дескрипторів, та програмної логіки, яка враховує контекст використання, політики підприємства і журнал попередніх підключень.

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням кількості кіберзагроз, які становлять суттєву небезпеку для конфіденційності, цілісності та доступності інформаційних ресурсів як окремих користувачів, так і підприємств різного масштабу. Серед різноманітних векторів атак особливе місце посідають загрози, пов'язані з використанням USB-інтерфейсів. Універсальна послідовна шина (Universal Serial Bus, USB) є найпоширенішим інтерфейсом підключення периферійних пристроїв до комп'ютерних систем, що одночасно робить її привабливим каналом для реалізації

кібератак [1, 2].

За даними звіту IBM X-Force Threat Intelligence Index 2025, середня вартість витоку даних, спричиненого через фізичні носії та USB-пристрої, становить 4,45 мільйона доларів США. Водночас, згідно з дослідженнями SANS Institute, понад 70% організацій стикалися з інцидентами безпеки, пов'язаними з неконтрольованим використанням USB-пристроїв. Ці факти свідчать про критичну необхідність розробки ефективних систем захисту від USB-загроз [3, 4].

Актуальність даного дослідження зумовлена появою нових класів атак, таких як BadUSB, Rubber Ducky, O.MG Cable та USB Killer, які використовують вразливості протоколу USB на апаратному рівні. Такі атаки здатні обійти традиційні програмні засоби захисту, оскільки шкідливі пристрої маскуються під легітимні периферійні пристрої, зокрема клавіатури або мережеві адаптери [5, 6].

Особливу загрозу становить людський фактор. За статистикою, близько 90% успішних кібератак мають людську складову як ключовий елемент. У контексті USB-загроз це проявляється через соціальну інженерію – підкидання заражених флеш-носіїв (USB Drop Attack), навмисне підключення шкідливих пристроїв інсайдерами та підключення невідомих пристроїв недостатньо обізнаним персоналом. Хоча повністю усунути загрозу людського фактора технічними засобами неможливо, розроблений у цій роботі комплекс максимально мінімізує її наслідки на технічному рівні [7, 8].

Існуючі комерційні рішення для контролю USB-пристроїв, такі як Symantec Endpoint Protection або McAfee Device Control, є переважно програмними та мають значну вартість ліцензування – від 40 доларів за одне робоче місце. При цьому вони не забезпечують апаратного рівня фільтрації, що залишає можливість обходу захисту на рівні USB-протоколу. Розробка власного апаратно-програмного комплексу дозволяє створити значно дешевше (вартість апаратної частини – від 15 доларів на базі Arduino), більш надійне та гнучке рішення, яке забезпечує захист на двох рівнях одночасно [9].

Крім фінансової вигоди, запропонований комплекс забезпечує відповідність вимогам міжнародних стандартів інформаційної безпеки, зокрема ISO/IEC 27001 та

NIST SP 800-53, які вимагають обов'язкового контролю знімних носіїв інформації та периферійних USB-пристроїв на підприємствах. Модульна архітектура системи дозволяє легко масштабувати рішення на все підприємство без значних додаткових витрат [10, 11].

Об'єктом дослідження є процес захисту комп'ютерних систем від загроз, що реалізуються через USB-інтерфейси.

Предметом дослідження є методи та засоби виявлення, запобігання та нейтралізації загроз, що реалізуються через USB-інтерфейси, з використанням апаратно-програмного комплексу.

Мета роботи полягає в дослідженні та розробці апаратно-програмного комплексу для виявлення та протидії загрозам USB-інтерфейсів, що забезпечує виявлення, запобігання та нейтралізацію загроз через USB-інтерфейси та підвищення рівня інформаційної безпеки.

Для досягнення поставленої мети визначено такі завдання: 1) провести аналіз сучасних загроз інформаційній безпеці, що реалізуються через USB-інтерфейси, та класифікувати існуючі типи USB-атак; 2) дослідити наявні засоби захисту від USB-загроз та визначити їх переваги та недоліки; 3) розробити модель загроз для USB-інтерфейсів з урахуванням усіх класів USB-пристроїв; 4) спроєктувати архітектуру двомодового апаратно-програмного комплексу захисту від USB-атак; 5) реалізувати апаратну частину комплексу на базі мікроконтролера з використанням мов C/C++; 6) реалізувати програмну частину комплексу мовою Python з підтримкою шифрованого логування; 7) провести тестування розробленого комплексу в різних сценаріях USB-атак та оцінити його ефективність.

Методи дослідження: аналіз та систематизація наукових джерел, метод моделювання загроз, метод порівняльного аналізу, експериментальне тестування, криптографічні методи захисту інформації.

Практичне значення роботи полягає у створенні готового до впровадження апаратно-програмного комплексу, що може бути використаний для захисту комп'ютерних систем малого та середнього бізнесу від USB-атак із мінімальними фінансовими витратами та максимальним рівнем безпеки.

1 АНАЛІЗ ЗАГРОЗ ІНФОРМАЦІЙНІЙ БЕЗПЕЦІ ЧЕРЕЗ USB-ІНТЕРФЕЙСИ

1.1 Класифікація та аналіз сучасних USB-загроз

Проблема захисту USB-інтерфейсів є актуальною для більшості сучасних комп'ютерних систем, оскільки цей інтерфейс поєднує зручність використання, універсальність і безпосередній доступ зовнішнього USB-пристрою до робочої станції. Через USB-порт до комп'ютера можуть підключатися знімні носії, клавіатури, маніпулятори, мобільні телефони, периферійне обладнання, програматори, мережеві адаптери та інші пристрої. Така універсальність створює не лише переваги, а й додаткову поверхню атаки, оскільки система часто автоматично розпізнає пристрій і надає йому певний рівень взаємодії з операційним середовищем.

У роботі розглянуто проблему, пов'язану з тим, що USB-пристрій не завжди є звичайним носієм інформації або периферійним елементом. Один і той самий фізичний пристрій може представлятися системі як клавіатура, накопичувач, мережевий адаптер або складений пристрій із кількома функціями. Якщо контроль підключень здійснюється лише за зовнішнім виглядом або назвою пристрою, користувач не має достатніх підстав для оцінювання його реальної поведінки. Саме тому в дипломній роботі увагу зосереджено на поєднанні програмного аналізу параметрів підключення та апаратної можливості блокування небезпечного каналу.

Проаналізовано основні загрози, які реалізуються через USB-інтерфейс. До них належать атаки типу BadUSB, HID-ін'єкція, USB Drop Attack, підміна дескрипторів, несанкціоноване копіювання інформації на знімні носії, витік даних через зовнішній USB-пристрій, а також фізичне пошкодження обладнання через некоректне або навмисно небезпечне електричне підключення. Кожен із цих сценаріїв відрізняється способом реалізації, однак спільною ознакою є використання довіри комп'ютерної системи до підключеного USB-пристрою.

BadUSB є одним із найбільш показових прикладів загрози, оскільки у цьому випадку пристрій змінює або імітує свою функціональність на рівні вбудованого програмного забезпечення контролера. Такий USB-пристрій може виглядати як

звичайний накопичувач, але після підключення поводитися як клавіатура, мережевий адаптер або інший клас обладнання. Проблема полягає в тому, що звичайна перевірка файлів на носії не дозволяє повністю оцінити поведінку самого пристрою. Тому система захисту має враховувати не лише наявність файлів, а й клас пристрою, його дескриптори, режим підключення та відповідність базі довірених пристроїв.

HID-ін'єкція пов'язана з імітацією клавіатури або іншого пристрою введення. У такому випадку зовнішній USB-пристрій може автоматично передавати послідовність натискань клавіш, запускати службові команди, відкривати системні вікна або виконувати інші дії від імені користувача. Цей тип атаки особливо небезпечний через те, що операційна система зазвичай довіряє пристроям класу HID і не завжди вимагає додаткового підтвердження для їх роботи. У межах запропонованого підходу такі пристрої повинні проходити окрему перевірку, а політики реагування мають передбачати тимчасове або повне блокування невідомих HID-підключень.

USB Drop Attack базується на соціальному та фізичному чиннику. Зловмисник залишає заражений або спеціально підготовлений USB-пристрій у місці, де користувач може знайти його та підключити до робочої станції. Такий сценарій не потребує складного проникнення в мережу, оскільки користувач сам створює умови для запуску інциденту безпеки. Для протидії цьому типу загрози важливим є контроль підключень на рівні робочого місця, ведення шифрованого журналювання подій та використання бази довірених пристроїв.

Підміна дескрипторів USB-пристрою є окремою проблемою, оскільки система ідентифікує пристрій за набором параметрів, які можуть бути змінені або імітовані. До таких параметрів належать ідентифікатори виробника та продукту, серійний номер, клас пристрою, підклас, протокол і текстові поля опису. Якщо система покладається лише на один параметр, наприклад назву пристрою, зловмисник може обійти перевірку. Тому доцільно аналізувати сукупність ознак і порівнювати їх із записами бази довірених пристроїв.

Знімні носії становлять ризик не лише через можливість занесення

шкідливого вмісту, а й через витік даних. Навіть якщо пристрій не виконує активних дій, він може використовуватися для копіювання службових документів, конфігураційних файлів або іншої інформації, що має обмежений доступ. У роботі враховано, що захист USB-інтерфейсів має передбачати не тільки блокування підозрілих пристроїв, а й формування події безпеки, збереження її в зашифрованому журналі та надання адміністратору інформації для подальшого аналізу.

Фізичне пошкодження обладнання через USB-інтерфейс також належить до суттєвих загроз. Воно може бути пов’язане з подачею некоректної напруги, перевантаженням лінії живлення або використанням спеціально підготовленого пристрою, здатного вивести з ладу порт чи елементи системної плати. Для зменшення такого ризику програмного контролю недостатньо, оскільки реакція має відбуватися на рівні апаратного каналу. Саме тому в роботі передбачено мікроконтролерний USB-фільтр, який може розмикати лінії даних і керувати живленням зовнішнього USB-пристрою.

Таким чином, сучасні USB-загрози мають комбінований характер і можуть поєднувати програмний, апаратний та організаційний аспекти. Для ефективної протидії недостатньо застосовувати лише заборону автозапуску або перевірку файлів на знімному носії. Необхідним є комплексний підхід, який передбачає контроль підключень, аналіз параметрів USB-пристрою, використання політик реагування, апаратне блокування небезпечного каналу та шифроване журналювання подій.

Важливо враховувати, що USB-інтерфейс використовується не лише для передавання файлів, а й для підключення пристроїв введення, мережевих адаптерів, засобів автентифікації та спеціалізованого обладнання. Через це однакова фізична форма роз’єму не означає однаковий рівень ризику. Для користувача флешнакопичувач, клавіатура та зарядний кабель можуть виглядати як звичайні аксесуари, однак для комп’ютерної системи вони створюють різні сценарії взаємодії. Саме тому класифікація USB-загроз має ґрунтуватися не на зовнішньому вигляді пристрою, а на його функціях і поведінці після підключення.

Окрему небезпеку становлять складені USB-пристрої, які одночасно містять кілька функціональних інтерфейсів. Наприклад, один пристрій може представлятися системі як накопичувач і як пристрій введення. Такий підхід може бути цілком легітимним для деяких типів обладнання, але він також може використовуватися для обходу простих правил контролю. Якщо система дозволяє пристрій лише тому, що він має функцію накопичувача, вона може не врахувати прихований або додатковий інтерфейс, здатний виконувати небажані дії.

Ризик витоку даних через знімні носії посилюється тим, що користувачі часто сприймають копіювання файлів як звичайну робочу операцію. У малих офісах, навчальних аудиторіях і лабораторіях знімні носії можуть використовуватися для перенесення документів, програмного забезпечення або звітних матеріалів. Без контролю підключень адміністратор не завжди може встановити, який саме пристрій використовувався, коли він був підключений і чи відповідав він дозволеній політиці.

Фізичний аспект USB-загроз також не можна ігнорувати. Навіть якщо операційна система має програмні обмеження, вони не завжди здатні запобігти пошкодженню порту або збою живлення. Підключення несправного чи навмисно підготовленого пристрою може створити перевантаження або інші небажані електричні умови. Тому запропонована система передбачає апаратний рівень контролю, який доповнює програмні механізми.

Рисунок 1.1 – Класифікація загроз для USB-інтерфейсів

Таблиця 1.1 – Порівняльна характеристика основних USB-загроз

Тип загрози	Механізм реалізації	Наслідки	Ознаки виявлення
BadUSB	Зміна прошивки або класу пристрою	Виконання команд, обхід антивірусу	HID-клас без очікуваного сценарію
USB Drop Attack	Підкидання зараженого носія	Запуск шкідливих файлів, витік даних	Невідомий серійний номер, новий власник
O.MG Cable	Прихований бездротовий модуль у кабелі	Перехоплення введення, дистанційне керування	Незвичний HID або мережевий профіль
USB Killer	Високовольтний імпульс у порт	Фізичне пошкодження обладнання	Потребує фізичного обмеження порту

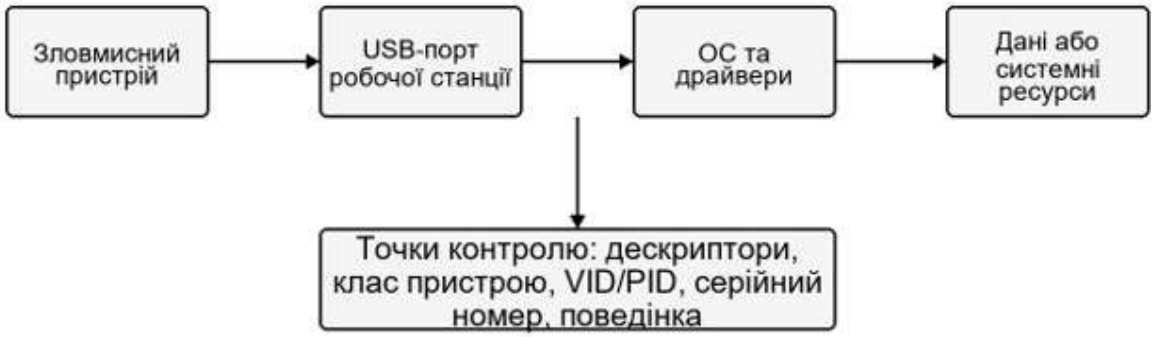


Рисунок 1.2 – Типовий канал реалізації атаки через USB-пристрій

Додаткову небезпеку створює те, що USB-атака часто маскується під звичайну робочу дію користувача. Підключення флеш-накопичувача, клавіатури, адаптера або кабелю саме по собі не викликає підозри, оскільки такі пристрої постійно використовуються в офісному та навчальному середовищі. Тому система захисту повинна оцінювати не лише сам факт підключення, а й набір ознак, які характеризують пристрій в момент ініціалізації. До таких ознак належать клас пристрою, заявлені інтерфейси, швидкість появи нових функцій, стабільність ідентифікаторів та відповідність очікуваному сценарію використання.

Окрему увагу доцільно приділяти комбінованим USB-пристроям, які одночасно можуть представлятися системі як накопичувач, клавіатура, мережевий адаптер або службовий інтерфейс. Саме така властивість робить USB зручним для

користувача, але водночас ускладнює контроль безпеки. Якщо засіб захисту перевіряє лише назву носія або серійний номер, він може пропустити прихований функціональний блок. Тому класифікація загроз повинна враховувати не тільки тип атаки, а й спосіб, у який пристрій взаємодіє з операційною системою після підключення.

1.2 Аналіз існуючих засобів захисту від USB-атак

Існуючі засоби захисту від USB-атак можна умовно поділити на організаційні, програмні та апаратні. Організаційні заходи включають правила використання знімних носіїв, інструктаж користувачів, обмеження фізичного доступу до робочих місць і регламенти реагування на інцидент безпеки. Такі заходи є необхідними, однак вони не усувають технічну можливість підключення небезпечного пристрою. Якщо користувач порушує вимоги або не розпізнає загрозу, самі правила не здатні забезпечити своєчасне блокування USB-інтерфейсу.

Програмні засоби контролю USB-пристроїв зазвичай працюють на рівні операційної системи. Вони можуть обмежувати доступ до знімних носіїв, забороняти запис, фіксувати факт підключення, перевіряти клас пристрою або застосовувати списки дозволених і заборонених пристроїв. Перевагою такого підходу є простота розгортання та можливість централізованого налаштування політик. Водночас програмний агент моніторингу починає діяти вже після того, як система розпізнала пристрій, тому в окремих випадках між моментом фізичного підключення та реакцією може виникати небажаний проміжок.

Засоби керування політиками операційної системи дозволяють обмежувати використання певних класів пристроїв або конкретних ідентифікаторів. Такий підхід є корисним у корпоративному середовищі, але має низку обмежень. Зокрема, підміна дескрипторів може ускладнювати точне визначення пристрою, а складені USB-пристрої можуть одночасно містити кілька функціональних інтерфейсів. Крім того, політика, налаштована занадто жорстко, може заважати нормальній роботі

користувача, а надто м’яка політика не забезпечує достатнього рівня контролю.

Апаратні засоби захисту можуть виконувати роль фізичного фільтра між комп’ютером і зовнішнім USB-пристроєм. Вони мають перевагу в тому, що можуть розірвати лінії живлення або даних незалежно від стану операційної системи. Такий підхід особливо важливий для протидії пристроям, які намагаються діяти одразу після підключення. Недоліком окремих апаратних рішень є обмежена гнучкість: без взаємодії з програмною частиною вони не завжди можуть враховувати контекст, політики користувача або базу довірених пристроїв.

Комерційні рішення для контролю USB-пристроїв часто орієнтовані на корпоративний сегмент і передбачають централізоване адміністрування, розширені політики доступу та звітність. Вони можуть бути ефективними для великих організацій, однак їх впровадження пов’язане з ліцензійними витратами, вимогами до інфраструктури та необхідністю супроводу. Для малого та середнього бізнесу, навчальних лабораторій або окремих робочих місць такі рішення не завжди є економічно доцільними.

Окрему групу становлять прості фізичні блокатори портів, які закривають USB-роз’єм або унеможливлюють підключення без спеціального ключа. Вони зменшують ризик несанкціонованого використання порту, але не вирішують задачу гнучкого контролю. Якщо користувачу все ж потрібно працювати з дозволеними пристроями, повне фізичне блокування створює незручності. Крім того, такий підхід не забезпечує шифроване журналювання подій і не дозволяє аналізувати спроби підключення.

У межах роботи запропоновано рішення, яке поєднує переваги програмного та апаратного підходів. Програмний агент моніторингу відповідає за контроль підключень, роботу з базою довірених пристроїв, застосування політик реагування та формування зашифрованого журналу подій. Мікроконтролерний USB-фільтр забезпечує фізичний рівень протидії, зокрема можливість розмикання ліній даних і керування живленням зовнішнього USB-пристрою.

Такий підхід дозволяє зменшити залежність від одного механізму захисту. Якщо програмна частина виявляє невідомий або потенційно небезпечний USB-

пристрій, вона передає команду апаратній частині, яка обмежує взаємодію пристрою з комп’ютером. Якщо ж пристрій входить до бази довірених пристроїв і відповідає встановленим параметрам, система може дозволити його роботу відповідно до політики. Це створює більш збалансовану модель захисту для прикладного використання.

Отже, аналіз існуючих засобів показує, що жоден окремий підхід не є достатнім для повного контролю USB-інтерфейсів. Організаційні заходи потребують технічної підтримки, програмні засоби залежать від стану операційної системи, а суто апаратні рішення часто не мають достатньої гнучкості. Тому доцільним є створення апаратно-програмного комплексу, який поєднує контроль підключень, політики реагування та фізичне блокування небезпечного каналу.

Перевагою вбудованих механізмів операційної системи є те, що вони не потребують окремого обладнання. Проте вони часто орієнтовані на адміністративне обмеження доступу, а не на повноцінну реакцію на інцидент безпеки. Наприклад, система може заборонити використання накопичувачів, але така заборона не завжди враховує пристрої іншого класу, які також можуть бути небезпечними. Через це захист має розглядатися ширше, ніж проста заборона запису на знімні носії.

Засоби антивірусного контролю корисні для перевірки файлів, але вони не вирішують проблему пристроїв, що діють не через файловий вміст. HID-ін’єкція або підміна дескрипторів можуть відбуватися без запуску підозрілого файлу з носія. Отже, перевірка вмісту є лише одним із допоміжних заходів і не може замінити контроль самого факту підключення та класу USB-пристрою.

У багатьох організаціях захист USB-портів зводиться до інструкцій для користувачів. Такий підхід має сенс як частина загальної політики безпеки, однак він залежить від дисципліни персоналу. Користувач може помилково підключити знайдений пристрій, використати особистий носій або не звернути увагу на незвичну поведінку комп’ютера після підключення. Тому організаційні правила мають підкріплюватися технічними механізмами.

Порівняно з повною фізичною забороною USB-портів запропонований

комплекс є більш гнучким. Він не вимагає відмови від усіх зовнішніх USB-пристроїв, а дозволяє працювати з довіреними пристроями та блокувати невідомі. Такий підхід краще відповідає реальним умовам експлуатації, де USB-інтерфейс часто залишається необхідним для виконання службових завдань.

Аналіз наявних засобів показує, що організаційні правила є необхідною, але недостатньою умовою захисту. Заборона використання невідомих носіїв, інструктаж користувачів і ведення журналів доступу знижують імовірність інциденту, однак не гарантують своєчасного блокування підозрілого пристрою. У реальних умовах користувач може помилитися, підключити знайдений носій або використати кабель невідомого походження. Через це організаційні заходи повинні доповнюватися технічними механізмами, які виконують перевірку незалежно від уважності користувача.

Програмні рішення зручні для централізованого адміністрування, але вони здебільшого працюють після того, як операційна система вже отримала інформацію про пристрій. Такий підхід є прийнятним для контролю доступу до файлів, проте може бути запізнілим для атак, що використовують швидку емуляцію клавіатури або зміну класу пристрою. Апаратні обмежувачі, навпаки, краще контролюють фізичний канал, але часто не мають достатньої гнучкості для аналізу контексту. Саме тому доцільним є поєднання програмного рішення з керованим апаратним модулем.

1.3 Огляд методів виявлення та нейтралізації USB-загроз

Методи виявлення USB-загроз мають враховувати особливості роботи USB-інтерфейсу, зокрема процедуру підключення, обмін дескрипторами та визначення класу пристрою. Після фізичного підключення комп’ютер отримує від пристрою набір параметрів, за якими визначає його тип і спосіб взаємодії. Саме на цьому етапі система захисту може отримати важливі ознаки для прийняття рішення. До таких ознак належать клас пристрою, ідентифікатори виробника та продукту,

серійний номер, опис інтерфейсів і наявність кількох функцій в одному пристрої.

Одним із базових методів є використання бази довірених пристроїв. У такій базі зберігаються параметри USB-пристроїв, яким дозволено працювати з конкретною робочою станцією. Якщо підключений пристрій відповідає запису в базі, система може надати доступ згідно з політикою. Якщо відповідність відсутня або параметри частково не збігаються, формується інцидент безпеки. Такий метод є простим для пояснення і практичного впровадження, однак він має застосовуватися разом із додатковою перевіркою класу та поведінки пристрою.

Для протидії підміні дескрипторів доцільно використовувати не один, а кілька параметрів ідентифікації. Наприклад, збіг лише назви пристрою не може вважатися достатнім, оскільки текстове поле може бути легко змінене. Більш надійним є зіставлення сукупності ознак: ідентифікаторів виробника та продукту, серійного номера, класу пристрою, кількості інтерфейсів і режиму роботи. Якщо один із параметрів змінюється без пояснюваної причини, система має розглядати таке підключення як потенційно небезпечне.

Окреме значення має контроль пристроїв класу HID. Такі пристрої зазвичай не потребують встановлення додаткових драйверів і можуть швидко почати передавати команди введення. Для зменшення ризику HID-ін’єкції система передбачає можливість блокування невідомих HID-пристроїв до моменту підтвердження або внесення до бази довірених пристроїв. У разі потреби адміністратор може створити окрему політику для службових клавіатур, сканерів штрихкодів або інших дозволених пристроїв цього класу.

Виявлення ризиків, пов’язаних зі знімними носіями, базується на контролі факту підключення, режиму доступу та відповідності пристрою встановленим правилам. Для таких пристроїв система може передбачати політики читання, заборону запису, тимчасове блокування або повну відмову в доступі. У роботі акцент зроблено не на перевірці конкретного вмісту носія, а на контролі самого підключення та можливості оперативного реагування, оскільки саме момент підключення є критичним для багатьох сценаріїв USB-атак.

Методи нейтралізації загроз мають бути пов’язані з рівнем ризику. Якщо

пристрій є довіреним, система може дозволити його роботу без додаткових дій. Якщо пристрій невідомий, але не має ознак критичної небезпеки, може застосовуватися режим очікування рішення користувача або адміністратора. Якщо пристрій належить до забороненого класу, має підозрілі дескриптори або викликає перевантаження лінії живлення, доцільним є апаратне блокування.

Апаратне блокування є важливим механізмом, оскільки воно дозволяє припинити взаємодію із зовнішнім USB-пристроєм незалежно від того, чи встигла операційна система завершити його ініціалізацію. Мікроконтролерний USB-фільтр може керувати лініями даних і живленням, виконуючи команди програмного агента моніторингу. Такий підхід дає можливість реалізувати не лише сповіщення про інцидент безпеки, а й безпосередню протидію.

Шифроване журналювання подій використовується для фіксації фактів підключення, прийнятих рішень і дій системи. Журнал має містити інформацію, необхідну для подальшого аналізу інциденту безпеки, але не повинен легко змінюватися або переглядатися сторонньою особою. У роботі передбачено, що події зберігаються в захищеному вигляді, а доступ до них має бути обмежений відповідними правами.

Таким чином, виявлення та нейтралізація USB-загроз у запропонованому підході базуються на поєднанні кількох механізмів: перевірки параметрів підключення, використання бази довірених пристроїв, аналізу класу пристрою, застосування політик реагування, апаратного блокування та шифрованого журналювання подій. Така комбінація дозволяє підвищити рівень контролю USB-інтерфейсів без надмірного ускладнення системи.

Виявлення загрози не повинно обмежуватися одноразовою перевіркою під час першого підключення. У практичній системі важливо фіксувати послідовність подій: коли пристрій з’явився, як він був ідентифікований, яку політику застосовано та який стан прийняв апаратний фільтр. Такий підхід дозволяє відновити картину інциденту безпеки після його виникнення і зрозуміти, чи була реакція системи коректною.

Для зменшення кількості помилкових рішень доцільно розділяти пристрої за

рівнем довіри. Довірений пристрій може працювати без додаткового втручання, умовно довірений може потребувати підтвердження, а невідомий або заборонений має блокуватися. Така градація дозволяє уникнути крайнощів, коли або блокується все, або дозволяється все без достатньої перевірки.

Політики реагування мають бути прозорими для адміністратора. Якщо система блокує пристрій, вона повинна пояснювати причину на рівні зрозумілого повідомлення: пристрій невідомий, клас заборонено, параметри не збігаються з базою довірених пристроїв або виникла помилка апаратного модуля. Це спрощує експлуатацію та допомагає швидше усунути неправильні налаштування.

Механізм нейтралізації через апаратний фільтр є особливо корисним у ситуаціях, коли рішення має виконуватися швидко. Якщо пристрій починає поводитися небажано одразу після підключення, простого запису події недостатньо. Система повинна мати засіб фізично обмежити подальшу взаємодію. Саме цю функцію виконує мікроконтролерний USB-фільтр у складі запропонованого комплексу.

Методи виявлення USB-загроз доцільно розглядати як послідовність етапів, а не як одну окрему перевірку. На першому етапі система фіксує сам факт підключення та отримує базові атрибути пристрою. На другому етапі виконується зіставлення цих атрибутів із базою довірених пристроїв і правилами політики безпеки. На третьому етапі оцінюється поведінка пристрою після підключення: чи з’являються додаткові інтерфейси, чи відбувається спроба швидкого введення команд, чи змінюється набір заявлених функцій.

Нейтралізація загрози має бути передбачуваною для адміністратора та мінімально шкідливою для користувача. Якщо пристрій явно не відповідає політиці, система повинна заблокувати канал обміну та зафіксувати інцидент у журналі. Якщо ознаки є неповними або суперечливими, доцільним є тимчасове обмеження доступу з подальшим запитом рішення адміністратора. Такий підхід дає змогу уникнути як надмірної довіри до невідомих пристроїв, так і необґрунтованого блокування легітимного обладнання.

Додатковим механізмом підвищення ефективності виявлення USB-загроз є

аналіз історії підключень пристроїв. Якщо один і той самий пристрій регулярно використовується на робочій станції та його параметри залишаються незмінними, рівень довіри до нього може бути підвищений. Навпаки, поява нового пристрою або зміна характеристик раніше відомого обладнання повинні розглядатися як подія, що потребує додаткової перевірки. Використання історичних даних дозволяє зменшити кількість помилкових спрацьовувань та підвищити точність прийняття рішень.

1.4 Постановка задачі

Розробка систем виявлення аномалій у логах комп'ютерних систем є складним і багатоетапним завданням, що поєднує методи кібербезпеки, машинного навчання, обробки послідовних даних і аналізу поведінкових шаблонів. У межах даної кваліфікаційної роботи передбачається вирішення комплексу взаємопов'язаних наукових і прикладних задач, спрямованих на створення ефективної інтелектуальної системи виявлення аномалій з використанням глибинної нейронної мережі типу LSTM-автоенкодера. З огляду на мету дослідження, формулюються наступні конкретні задачі:

- вивчення предметної області та аналіз сучасного стану проблеми;
- вивчення предметної області та аналіз сучасного стану проблеми;
- обґрунтування доцільності використання вибраної архітектури нейронної мережі, враховуючи специфіку вхідних даних;
- розробка методології попередньої обробки лог-даних. Потрібно розробити процедури для збору, очищення, нормалізації та парсингу логів із різних джерел, таких як системні журнали, журнали аудиту та мережеві логи. Здійснюється перетворення неструктурованих або напівструктурованих записів у формалізовану матрицю ознак, придатну для подачі у нейронну мережу;
- формалізація навчальної вибірки для моделі. Слід здійснити сегментацію даних у часові вікна із фіксованою довжиною, які формують послідовності входів

для LSTM-моделі. На цьому етапі визначається структура входу моделі, яка враховує кількість часових кроків, кількість ознак та формат представлення даних;

– проєктування та реалізація архітектури LSTM-автоенкодера. Необхідно розробити модель глибокої нейронної мережі, що включає шари LSTM-енкодера, латентне представлення, шари декодера та вихідний шар, який генерує реконструйовану послідовність. Підбираються функція втрат, оптимізатор, активаційні функції, кількість нейронів і шарів з урахуванням обчислювальної складності та здатності до узагальнення;

– налаштування та оптимізація гіперпараметрів моделі. Потрібно визначити оптимальні значення гіперпараметрів, таких як кількість нейронів, розмірність латентного простору, кількість епох, розмір пакету, швидкість навчання, коефіцієнт регуляризації тощо. Оптимізація може здійснюватися із використанням методів типу Hyperband, Grid Search або крос-валідації;

– навчання моделі на реальних лог-даних. Проводиться навчання автоенкодера на сукупності нормальних логів із репрезентативних наборів даних, зокрема BETH та Cisco Audit. У ході навчання модель формує уявлення про типову поведінку, що дозволяє використовувати її для виявлення аномалій шляхом аналізу похибки реконструкції;

– встановлення порогових значень аномалії та обчислення метрик. На основі результатів реконструкції визначається порогове значення, перевищення якого інтерпретується як аномалія. Проводиться розрахунок ключових метрик ефективності моделі: точності (Accuracy), повноти (Recall), точності позитивного передбачення (Precision), F-міри (F1-score), хибнопозитивної (FPR) та хибнонегативної (FNR) частот;

– оцінка достовірності та стійкості моделі. Модель тестується на відокремленому наборі даних, що містить як нормальні, так і аномальні записи, з метою перевірки її здатності до генералізації. Також перевіряється стабільність результатів на різних підмножинах даних за допомогою k-fold крос-валідації;

– формулювання висновків та перспектив подальших досліджень. Завершальним етапом є інтерпретація отриманих результатів, формулювання

висновків щодо ефективності розробленого підходу, а також виявлення можливих напрямів для подальшої оптимізації та масштабування системи, зокрема в напрямку потокової обробки логів або гібридних архітектур.

Постановка задачі в межах роботи передбачає не лише створення окремого засобу блокування, а й формування цілісного підходу до контролю USB-інтерфейсів. Система повинна виявляти нові підключення, перевіряти пристрій за набором технічних ознак, ухвалювати рішення відповідно до політики безпеки та виконувати дію реагування. Важливо, щоб кожен етап був відокремлений логічно, оскільки це спрощує тестування комплексу та подальше розширення його функцій.

До функціональних вимог належать підтримка бази довірених пристроїв, фіксація подій у журналі, можливість блокування невідомого пристрою та взаємодія з апаратним фільтром. До нефункціональних вимог належать простота розгортання, зрозумілість конфігурації, стійкість до помилок підключення та можливість роботи на звичайній робочій станції без складної інфраструктури. Такі вимоги відповідають практичній спрямованості бакалаврської роботи та дозволяють оцінити результат не лише теоретично, а й експериментально. Окрему увагу необхідно приділити можливості перевірки працездатності системи в умовах, наближених до реального використання. Для цього в межах роботи передбачається моделювання типових сценаріїв підключення USB-пристроїв, зокрема довіреного пристрою, невідомого пристрою та пристрою, що потребує блокування. Це дозволяє оцінити коректність прийняття рішень і підтвердити практичну придатність запропонованого комплексу.

2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ЗАХИСТУ ВІД USB-АТАК

2.1 Модель загроз та принципи захисту від USB-загроз

Проєктування апаратно-програмного комплексу починається з визначення моделі загроз, оскільки саме вона задає вимоги до структури системи, її функцій і механізмів реагування. У межах цієї роботи розглядається робоча станція, до якої користувач може підключати зовнішні USB-пристрої. Загроза виникає тоді, коли пристрій є невідомим, навмисно модифікованим, підробленим або використовується з порушенням встановлених правил безпеки.

Модель загроз передбачає, що зловмисник може мати фізичний доступ до USB-порту або може передати користувачу зовнішній USB-пристрій під виглядом звичайного носія інформації чи периферійного обладнання. У такому сценарії небезпека полягає не лише у вмісті знімного носія, а й у поведінці самого пристрою. Він може імітувати клавіатуру, змінювати дескриптори, намагатися створити новий мережевий інтерфейс або викликати перевантаження лінії живлення.

У роботі запропоновано принцип багаторівневого контролю. Перший рівень пов’язаний із програмним агентом моніторингу, який аналізує параметри підключення та порівнює USB-пристрій із базою довірених пристроїв. Другий рівень пов’язаний із політиками реагування, які визначають допустимі дії для різних класів пристроїв і сценаріїв. Третій рівень реалізується апаратною частиною, яка може фізично обмежити взаємодію пристрою з комп’ютером.

Основним принципом захисту є недовіра до невідомого підключення. Якщо зовнішній USB-пристрій не внесений до бази довірених пристроїв або має параметри, що не відповідають очікуваним, система не повинна автоматично дозволяти його повну роботу. У таких випадках формується інцидент безпеки, записується подія, а подальша дія визначається політикою реагування. Це може бути блокування, тимчасове очікування рішення або дозвіл лише для обмеженого режиму роботи [потрібно уточнити].

Для пристроїв класу HID у моделі загроз передбачено підвищений рівень

уваги. Такий пристрій може виконувати введення команд без участі користувача, тому невідомі HID-підключення мають перевірятися окремо. Якщо пристрій не є довіреним, система повинна мати можливість оперативно заблокувати лінії даних або живлення через мікроконтролерний USB-фільтр.

Для знімних носіїв важливо враховувати ризик витоку даних. Навіть якщо пристрій не виконує активних команд, він може використовуватися для копіювання інформації з робочої станції. Тому політики реагування мають передбачати не лише повне блокування, а й режими, пов’язані з дозволом читання, забороною запису або фіксацією факту підключення для подальшого розгляду [потрібно уточнити].

Апаратний рівень захисту в запропонованій моделі виконує роль незалежного виконавчого механізму. Якщо програмний агент моніторингу приймає рішення про блокування, мікроконтролерний USB-фільтр повинен розімкнути відповідні лінії або вимкнути живлення зовнішнього USB-пристрою. Це дозволяє не обмежуватися повідомленням користувачу, а виконувати практичну протидію загрозі.

Шифроване журналювання подій є допоміжним, але важливим принципом захисту. Воно дозволяє зберігати інформацію про підключення, рішення системи, спроби використання невідомих пристроїв і спрацювання політик реагування. Наявність такого журналу підвищує прозорість роботи системи та полегшує аналіз інцидентів безпеки після їх виникнення.

Таким чином, модель загроз для розроблюваного комплексу базується на реалістичному припущенні, що небезпечний пристрій може бути підключений без попередження, а реакція має бути швидкою, зрозумілою та технічно забезпеченою. Саме тому система передбачає поєднання програмного контролю, апаратного блокування, бази довірених пристроїв, політик реагування та захищеного збереження подій.

У межах моделі загроз користувач не завжди розглядається як зловмисник. Часто інцидент безпеки виникає через необережність, недостатню обізнаність або звичку підключати будь-який знайдений носій. Тому система має бути розрахована не лише на навмисні дії, а й на помилки користувача. Це особливо важливо для

навчальних лабораторій та невеликих організацій, де централізований контроль може бути обмеженим.

Модель загроз також враховує, що небезпечний USB-пристрій може не мати помітних зовнішніх ознак. Він може виглядати як звичайна флешка, кабель або адаптер. Через це рішення не повинно покладатися на візуальний контроль. Основою оцінювання мають бути параметри, отримані під час підключення, і реакція системи відповідно до встановлених політик.

Принцип мінімальної довіри до нового підключення дозволяє зменшити ризик автоматичного прийняття небезпечного пристрою як дозволеного. Якщо пристрій підключається вперше, система має розглядати його як невідомий до моменту перевірки. Це не означає, що кожне підключення є атакою, але означає, що доступ не повинен надаватися без контролю.

Ще одним принципом є розділення рішення та виконання. Програмний агент аналізує ситуацію і визначає дію, а апаратний фільтр фізично виконує блокування або дозвіл. Таке розділення підвищує керованість комплексу і дозволяє надалі змінювати політики без повної зміни апаратної частини.

Під час побудови моделі загроз важливо враховувати, що зловмисник не обов’язково має тривалий доступ до комп’ютера. У багатьох сценаріях достатньо короткочасного фізичного підключення пристрою, після чого шкідлива дія виконується автоматично. Це означає, що система захисту повинна реагувати на ранній стадії підключення, коли пристрій ще не отримав повноцінного доступу до функцій операційної системи. Саме раннє виявлення дає змогу зменшити ризик виконання команд або передачі даних.

Для опису ризиків доцільно використовувати групування загроз за характером впливу: підміна пристрою, несанкціоноване введення команд, витік інформації, пошкодження обладнання та приховування факту інциденту. Така класифікація узгоджується з логікою STRIDE, оскільки дозволяє пов’язати конкретну технічну подію з наслідками для конфіденційності, цілісності та доступності. У результаті модель загроз стає основою для вибору політик реагування, а не лише описом можливих атак.

Принцип нульової довіри до USB-пристроїв у цьому контексті означає, що новий пристрій не отримує дозвіл лише через наявність коректних дескрипторів або знайомої назви. Довіра повинна підтверджуватися сукупністю ознак і попередньо визначеними правилами. Якщо хоча б один важливий параметр не відповідає очікуваному профілю, система має перейти до обмеженого режиму роботи. Такий підхід робить захист більш стійким до простого копіювання зовнішніх ідентифікаторів пристрою.

2.2 Проєктування апаратної частини комплексу

Апаратна частина комплексу призначена для фізичного контролю каналу підключення зовнішнього USB-пристрою. Її основне завдання полягає у тому, щоб забезпечити можливість дозволу або блокування взаємодії між комп’ютером і пристроєм не лише на рівні операційної системи, а й на рівні електричних ліній. Такий підхід є важливим для протидії USB-загрозам, які можуть проявлятися одразу після підключення або не залежать від файлового вмісту носія.

Основою апаратної частини є мікроконтролерний USB-фільтр. Він приймає команди від програмного агента моніторингу, керує виконавчими елементами та передає інформацію про стан системи. Мікроконтролер виконує роль проміжного елемента між програмним рішенням і фізичним каналом USB. Завдяки цьому програмний агент може не тільки формувати повідомлення про інцидент безпеки, а й ініціювати реальне блокування зовнішнього USB-пристрою.

У структурі апаратної частини передбачено USB-роз’єм для підключення до комп’ютера, USB-роз’єм для зовнішнього пристрою, елементи захисту ліній даних, керований ключ живлення, комутатор ліній D+ і D-, а також службовий канал зв’язку з мікроконтролером. Такий склад дозволяє контролювати як інформаційний обмін, так і подачу живлення на зовнішній USB-пристрій.

Для захисту ліній даних доцільно передбачити TVS/ESD-елементи біля USB-роз’ємів. Вони зменшують ризик пошкодження обладнання під час

електростатичного розряду або некоректного підключення. Окрім цього, у схемі може застосовуватися синфазний дросель для ліній D+ і D-, що зменшує вплив завад і покращує стабільність передавання сигналу. Конкретні номінали та типи компонентів мають бути уточнені під час остаточного розроблення схеми [потрібно уточнити].

Керований ключ живлення використовується для подачі або відключення напруги на downstream USB-порт. Він дозволяє апаратно вимкнути зовнішній USB-пристрій у разі порушення політики або виявлення небезпечного підключення. Додатково такий ключ може формувати сигнал про перевантаження або коротке замикання, що передається на мікроконтролер як ознака потенційного інциденту безпеки.

Комутатор ліній даних використовується для керованого розмикання або з'єднання ліній D+ і D-. Якщо пристрій є довіреним, комутатор перебуває у стані дозволу, і зовнішній USB-пристрій може взаємодіяти з комп'ютером. Якщо пристрій невідомий або заблокований політикою, комутатор розриває канал обміну. Такий механізм є важливим для протидії HID-ін'єкції та іншим сценаріям, де небезпечні дії можуть виконуватися через лінії даних.

Для контролю споживання струму може застосовуватися датчик струму, підключений до мікроконтролера через I2C. Його використання дозволяє оцінювати стан живлення зовнішнього USB-пристрою та виявляти ситуації, пов'язані з перевантаженням або нестандартною поведінкою. У межах роботи цей елемент розглядається як допоміжний засіб підвищення інформативності системи, а не як єдиний критерій небезпеки.

Апаратна частина також передбачає індикацію стану. Зелений світлодіод може використовуватися для позначення дозволеного підключення, червоний — для блокування або інциденту безпеки, а звуковий сигнал — для привернення уваги користувача. Дисплей або простий індикаторний модуль може відображати короткий стан системи, однак його використання залежить від обраної конструкції [потрібно уточнити].

Проєктування апаратної частини має враховувати практичну зручність

використання. Модуль повинен розміщуватися між комп’ютером і зовнішнім USB-пристроєм, не ускладнюючи роботу з дозволеними пристроями. Водночас усі критичні елементи, пов’язані із захистом ліній і комутацією живлення, мають бути розташовані так, щоб забезпечувати надійне розмикання каналу в разі спрацювання політики реагування.

Під час проєктування апаратної частини важливо забезпечити коректне розмежування службового каналу керування та каналу підключення зовнішнього USB-пристрою. Службовий канал використовується для обміну командами між програмним агентом і мікроконтролером, тоді як основний канал призначений для роботи зовнішнього пристрою. Таке розмежування спрощує логіку схеми та зменшує ймовірність помилок у керуванні.

Живлення апаратного модуля має бути організоване так, щоб керуюча частина залишалася працездатною навіть у випадку блокування downstream-порту. Якщо вимкнення живлення зовнішнього USB-пристрою одночасно вимикатиме мікроконтролер, система втратить можливість відновлення контролю. Тому живлення логіки та живлення зовнішнього порту доцільно розглядати як окремі ділянки схеми.

Для апаратного блокування ліній даних потрібно обирати комутатор, придатний для роботи з USB-сигналами. У пояснювальній записці достатньо показати принципове рішення, однак під час практичного виготовлення необхідно уточнити конкретну мікросхему, її електричні параметри та допустимий режим роботи [потрібно уточнити]. Це дозволить уникнути ситуації, коли схема формально правильна, але не відповідає вимогам сигналів USB.

Конструктивно апаратний модуль має бути зручним для підключення. Вхідний порт до комп’ютера і вихідний порт до зовнішнього USB-пристрою бажано розташовувати так, щоб користувач не плутав напрямок підключення. Додаткова індикація стану дозволяє швидко зрозуміти, чи порт відкритий, заблокований або очікує рішення.

Апаратна частина комплексу повинна виконувати роль керованої межі між комп’ютером і зовнішнім USB-пристроєм. Її призначення полягає не в повній

заміні програмного аналізу, а у забезпеченні фізичної можливості швидко припинити обмін даними або живленням. У такій архітектурі апаратний модуль діє як виконавчий елемент, тоді як програмний агент визначає політику та приймає рішення. Це розмежування спрощує логіку системи й дозволяє окремо перевіряти програмні та апаратні функції.

Під час проєктування USB-фільтра необхідно враховувати електричні та експлуатаційні обмеження. Модуль не повинен створювати нестабільне з'єднання для дозволених пристроїв, а також має коректно переходити у безпечний стан у разі помилки керування. Безпечним станом доцільно вважати такий режим, у якому невідомий пристрій не може вільно взаємодіяти з комп'ютером. Це особливо важливо, якщо програмний агент тимчасово недоступний або канал зв'язку з мікроконтролером перерваний.

Окремим проєктним питанням є передача команд між програмним агентом і мікроконтролером. Команди мають бути простими, однозначними та придатними для журналювання. Наприклад, програмна частина може передавати мікроконтролеру рішення дозволити, обмежити або заблокувати пристрій, а апаратна частина повертає стан виконання. Такий мінімальний протокол зменшує ймовірність помилок реалізації та робить поведінку комплексу зрозумілою під час тестування.

2.3 Проєктування програмної частини комплексу

Програмна частина комплексу виконує функції моніторингу, прийняття рішення та взаємодії з апаратним модулем. Вона має працювати як програмний агент моніторингу, який фіксує підключення USB-пристрою, отримує його параметри, перевіряє відповідність базі довірених пристроїв і застосовує політики реагування. Саме програмна частина забезпечує логіку роботи системи та формує зрозумілий для користувача або адміністратора результат.

Першим завданням програмного агента є контроль підключень. Після появи

нового зовнішнього USB-пристрою агент має отримати доступні параметри пристрою: клас, ідентифікатори, серійний номер, опис і режим роботи. На основі цих даних формується запис події, який надалі використовується для перевірки пристрою та збереження інформації в зашифрованому журналі.

База довірених пристроїв є одним із ключових елементів програмної частини. Вона містить записи про пристрої, яким дозволено працювати з конкретною системою. До запису можуть входити ідентифікатори пристрою, його клас, серійний номер, опис і дозволений режим роботи. Якщо пристрій відповідає запису в базі, система може дозволити його використання. Якщо відповідність не знайдена, агент застосовує політику для невідомих пристроїв.

Політики реагування визначають, які дії має виконати система у відповідь на певну подію. Для довірених пристроїв може застосовуватися дозвіл роботи, для невідомих знімних носіїв — тимчасове блокування або запит підтвердження, для невідомих HID-пристроїв — негайне блокування. Конкретний набір політик може бути змінений відповідно до умов експлуатації [потрібно уточнити]. Важливо, щоб політики були зрозумілими та не створювали зайвої складності для адміністратора.

Модуль взаємодії з апаратною частиною відповідає за передавання команд мікроконтролерному USB-фільтру. Якщо пристрій дозволено, агент надсилає команду відкриття каналу або подачі живлення. Якщо пристрій заблоковано, агент передає команду розмикання ліній даних або відключення живлення. Таким чином, програмне рішення безпосередньо впливає на фізичний стан USB-підключення.

Шифроване журналювання подій необхідне для збереження інформації про роботу системи. До журналу можуть вноситися дата і час події, параметри USB-пристрою, результат перевірки, застосована політика та дія апаратного модуля. Захищений формат збереження потрібний для того, щоб сторонній користувач не міг легко змінити або приховати інформацію про інцидент безпеки.

Графічний інтерфейс програмного агента має забезпечувати базові дії: перегляд стану підключень, відображення останніх подій, роботу з базою довірених пристроїв і налаштування політик реагування. Інтерфейс не повинен бути перевантаженим, оскільки основним завданням є оперативне розуміння стану

USB-порту та швидке прийняття рішення. Для бакалаврської роботи достатньо реалізувати інтерфейс, який демонструє основну логіку комплексу та дозволяє виконувати ключові дії.

Програмна частина має враховувати помилки та нестандартні ситуації. Наприклад, апаратний модуль може бути тимчасово недоступним, пристрій може бути від’єднаний під час перевірки, а дані про дескриптори можуть бути неповними. У таких випадках агент повинен переходити до безпечного режиму, фіксувати подію та не надавати невідомому пристрою необмежений доступ без перевірки.

Таким чином, програмна частина комплексу є центром прийняття рішень. Вона поєднує контроль підключень, базу довірених пристроїв, політики реагування, шифроване журналювання подій і керування апаратним фільтром. Саме через цю взаємодію система перетворюється з простого засобу спостереження на прикладний комплекс протидії USB-загрозам.

Програмний агент має працювати так, щоб користувач не міг випадково обійти основні правила контролю. Якщо пристрій невідомий, рішення про дозвіл має прийматися відповідно до політики, а не лише через механічне підтвердження без пояснення. Для цього інтерфейс повинен показувати основні параметри пристрою та причину спрацювання правила.

База довірених пристроїв потребує механізму додавання нових записів. Доцільно, щоб додавання виконувалося після контрольованого підключення пристрою і підтвердження адміністратором. У записі мають зберігатися ті параметри, які система реально може отримати. Якщо певний параметр відсутній, це потрібно враховувати в політиці, а не вигадувати значення.

Політики реагування повинні бути збережені у зрозумілому форматі, щоб їх можна було переглядати та змінювати без втручання в програмний код. Для дипломної реалізації достатньо передбачити базовий набір правил, однак сама структура має дозволяти подальше розширення. Це відповідає прикладному характеру роботи та показує можливість розвитку системи.

Шифроване журналювання подій має виконуватися автоматично.

Користувач не повинен вручну створювати записи про підключення, оскільки це знижує надійність фіксації інцидентів. Кожна значуща дія системи повинна супроводжуватися записом, який у подальшому може бути переглянутий уповноваженою особою.

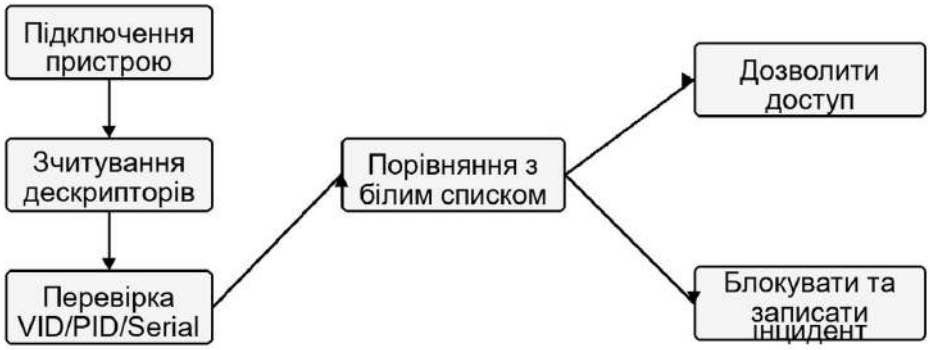


Рисунок 2.1 – Алгоритм перевірки USB-пристрою

Рисунок 2.2 – Логічна структура бази даних системи

Таблиця 2.1 – Модулі програмного агента

Модуль	Функції	Ключові дані
1	2	3
mcu_auth	Автентифікація апаратного фільтра	challenge, HMAC, ключ

Кінець таблиці 2.1

1	2	3
usb_monitor	Отримання подій підключення	VID, PID, serial, class
policy_engine	Прийняття рішення	whitelist, policies, risk_score
Logger	Шифроване журналювання	timestamp, action, reason
Blocker	Нейтралізація пристрою	port_id, device_id, action

Програмна частина має будуватися за модульною структурою, оскільки різні функції комплексу мають різну природу. Модуль моніторингу відповідає за отримання подій від операційної системи, модуль аналізу порівнює параметри пристрою з правилами, модуль реагування формує команду для апаратного фільтра, а модуль журналювання зберігає результати перевірки. Завдяки такому поділу зміна одного компонента не потребує повної переробки всієї системи.

База довірених пристроїв повинна містити не лише назву пристрою, а й набір ознак, які важче підмінити випадково. Доцільно зберігати ідентифікатори виробника та продукту, серійний номер, клас пристрою, допустимі інтерфейси, дату внесення до бази та примітку адміністратора. Наявність кількох атрибутів дозволяє гнучкіше оцінювати пристрій і виявляти ситуації, коли зовні схожий об’єкт має інший функціональний профіль.

Програмний агент також повинен враховувати помилки та нестандартні ситуації. Пристрій може бути від’єднаний під час перевірки, операційна система може надати неповний набір параметрів, а апаратний модуль може не відповісти на команду з першої спроби. У таких випадках система не повинна завершувати роботу аварійно. Коректна обробка винятків, повторна спроба зв’язку та запис події до журналу підвищують практичну придатність комплексу.

2.4 Проєктування архітектури та взаємодії компонентів

Архітектура комплексу побудована за модульним принципом, що дозволяє розділити функції виявлення, прийняття рішення та фізичного реагування. Такий підхід спрощує розроблення, тестування та подальше вдосконалення системи. Основними складовими є програмний агент моніторингу, база довірених пристроїв, модуль політик реагування, механізм шифрованого журналювання подій і мікроконтролерний USB-фільтр.

Програмний агент моніторингу працює на робочій станції та відповідає за взаємодію з операційним середовищем. Він отримує інформацію про підключені USB-пристрої, порівнює її з базою довірених пристроїв і визначає, чи відповідає підключення встановленим правилам. Результат перевірки передається до модуля політик реагування, який формує подальшу дію.

База довірених пристроїв виконує роль сховища дозволених USB-пристроїв і параметрів їх використання. Вона має бути достатньо простою для адміністрування, але водночас містити ті ознаки, які дозволяють зменшити ризик підміни дескрипторів. Для цього доцільно зберігати не лише назву пристрою, а й технічні параметри, що отримуються під час підключення.

Модуль політик реагування забезпечує перехід від факту виявлення підключення до конкретної дії. Якщо пристрій довірений, політика дозволяє його роботу. Якщо пристрій невідомий або належить до небажаного класу, політика може ініціювати блокування, запит підтвердження або запис події без надання доступу. Завдяки цьому система може адаптуватися до різних умов використання.

Мікроконтролерний USB-фільтр є виконавчим елементом архітектури. Він отримує команди від програмного агента та керує станом ліній даних і живлення. У випадку дозволу він замикає канал взаємодії між комп’ютером і зовнішнім USB-пристроєм. У випадку блокування він розмикає канал або відключає живлення, що зменшує ризик виконання небезпечних дій.

Взаємодія компонентів відбувається за послідовною логікою. Спочатку користувач підключає USB-пристрій до апаратного модуля. Далі програмний агент отримує інформацію про підключення, перевіряє її за базою довірених пристроїв і

політиками реагування. Після цього агент передає команду мікроконтролеру, а той змінює стан апаратного каналу. Паралельно формується запис у зашифрованому журналі подій.

Для зменшення ризику помилкового дозволу система має працювати за принципом безпечного стану. Якщо неможливо однозначно визначити пристрій, якщо апаратний модуль не відповідає або якщо політика не налаштована, перевага має надаватися обмеженню доступу. Такий підхід є доцільним для системи захисту, оскільки невідоме підключення не повинно автоматично отримувати довіру.

Архітектура також передбачає можливість розширення. До системи можна додати нові політики реагування, додаткові параметри перевірки пристроїв, інші способи індикації або централізоване збереження подій [потрібно уточнити]. Водночас базова структура залишається зрозумілою: програмна частина аналізує та приймає рішення, апаратна частина виконує фізичну дію.

Запропонована архітектура відповідає прикладному характеру роботи, оскільки орієнтована на створення працездатного комплексу, а не лише на опис окремих загроз. Її практичне значення полягає у тому, що вона дозволяє поєднати контроль підключень, реагування на інциденти безпеки та фізичний вплив на USB-канал у межах одного рішення.

Модульність архітектури дає змогу окремо перевіряти програмну та апаратну частини. Наприклад, програмний агент можна тестувати з імітацією відповіді апаратного модуля, а мікроконтролерний фільтр — окремо перевіряти на виконання команд дозволу та блокування. Це спрощує розроблення і дозволяє швидше знаходити помилки.

У взаємодії компонентів важливо передбачити підтвердження виконання команд. Якщо програмний агент передав команду блокування, він повинен отримати від апаратної частини відповідь про зміну стану або повідомлення про помилку. Без такого підтвердження система не може впевнено вважати, що небезпечний канал справді розірвано.

Архітектура має враховувати ситуації, коли один із компонентів тимчасово недоступний. Якщо не працює база довірених пристроїв, якщо пошкоджено файл

налаштувань або якщо мікроконтролерний USB-фільтр не відповідає, система повинна переходити до обмежувального режиму. Це відповідає принципу безпечного стану.

З погляду практичного застосування важливо, щоб адміністратор міг пояснити роботу системи користувачу. Користувач має розуміти, що блокування відбувається не випадково, а через невідповідність пристрою встановленим правилам. Це зменшує кількість конфліктних ситуацій і підвищує дисципліну використання USB-інтерфейсів.

2.5 Висновки до розділу

У другому розділі розглянуто проєктування апаратно-програмного комплексу для виявлення та протидії загрозам, що реалізуються через USB-інтерфейси. На основі проаналізованих сценаріїв сформовано модель загроз, у якій небезпечним вважається невідоме, підроблене або неконтрольоване підключення зовнішнього USB-пристрою.

Запропоновано архітектуру комплексу, що включає програмний агент моніторингу, базу довірених пристроїв, політики реагування, шифроване журналювання подій і мікроконтролерний USB-фільтр. Така структура дозволяє розділити функції аналізу, прийняття рішення та фізичного блокування USB-каналу.

Проєктування апаратної частини передбачає використання керованого ключа живлення, комутатора ліній даних, елементів захисту USB-ліній, індикації стану та мікроконтролера. Це забезпечує можливість практичного реагування на інцидент безпеки шляхом розмикання ліній або відключення живлення зовнішнього USB-пристрою.

Проєктування програмної частини передбачає контроль підключень, перевірку параметрів пристроїв, застосування бази довірених пристроїв, виконання політик реагування та збереження подій у захищеному вигляді. Такий підхід

дозволяє не обмежуватися пасивним спостереженням, а формувати керовану реакцію на потенційно небезпечні USB-підключення.

Отримані проєктні рішення є основою для подальшої реалізації комплексу. Вони визначають склад апаратної та програмної частин, принципи їх взаємодії та очікуваний результат роботи системи в умовах підключення довірених, невідомих або потенційно небезпечних USB-пристроїв.

У розділі також обґрунтовано необхідність апаратного рівня протидії. Програмна перевірка є корисною, але для частини USB-загроз потрібна можливість фізичного розриву каналу. Саме тому в структурі комплексу передбачено мікроконтролерний USB-фільтр, який виконує команди програмного агента.

Запропоновані рішення залишають можливість подальшого розвитку. За потреби комплекс може бути доповнений іншими політиками реагування, розширеною базою довірених пристроїв або додатковою індикацією стану [потрібно уточнити]. Проте базова модель уже охоплює ключові функції, необхідні для прикладного контролю USB-підключень.

Архітектура взаємодії компонентів повинна забезпечувати зрозумілий життєвий цикл USB-пристрою у системі контролю. Після появи пристрою формується подія підключення, далі виконується збір атрибутів, перевірка за правилами, вибір дії та передача команди на апаратний модуль. Завершальним етапом є фіксація результату в журналі. Така послідовність дозволяє відтворити хід прийняття рішення під час аналізу інциденту.

Важливою властивістю архітектури є відсутність єдиної точки, від якої повністю залежить захист. Якщо програмна частина не може одразу класифікувати пристрій, апаратний модуль має зберігати обмежений або заблокований стан. Якщо апаратна частина тимчасово недоступна, програмний агент повинен зафіксувати помилку та не вважати пристрій автоматично довіреним. Такий підхід відповідає принципу ешелонованого захисту та зменшує наслідки окремої відмови.

3 РОЗРОБКА ТА ТЕСТУВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ЗАХИСТУ ВІД USB-АТАК

3.1 Реалізація апаратної частини комплексу

Реалізація апаратної частини комплексу спрямована на створення керованого USB-фільтра, який може виконувати фізичне обмеження доступу зовнішнього USB-пристрою до комп’ютера. На відміну від суто програмного контролю, апаратний модуль здатний впливати на лінії даних і живлення, що є важливим для протидії загрозам типу BadUSB, HID-ін’єкції та фізичного пошкодження обладнання.

У практичній структурі апаратний модуль розміщується між робочою станцією та зовнішнім USB-пристроєм. Така схема дозволяє контролювати весь канал взаємодії. Вхідний роз’єм підключається до комп’ютера, вихідний роз’єм — до пристрою, а мікроконтролер керує виконавчими елементами відповідно до команд програмного агента моніторингу. Це створює зрозумілу модель роботи: пристрій не отримує повноцінного доступу, доки система не дозволить підключення.

Лінії D+ і D- проходять через елементи захисту та керований комутатор. Елементи захисту зменшують ризик пошкодження від електростатичних впливів, а комутатор дозволяє розімкнути канал даних у разі блокування пристрою. Таке рішення особливо важливе для невідомих HID-пристроїв, оскільки саме через лінії даних вони можуть передавати команди введення.

Лінія живлення VBUS проходить через самовідновний запобіжник і керований ключ живлення. Запобіжник зменшує ризик пошкодження внаслідок перевантаження, а ключ дозволяє вимкнути живлення зовнішнього USB-пристрою за командою мікроконтролера. У разі виявлення перевантаження або іншої нестандартної ситуації формується сигнал, який може бути переданий програмній частині як ознака інциденту безпеки.

Мікроконтролер виконує роль керуючого елемента. Він отримує команди дозволу або блокування, змінює стан ключа живлення та комутатора ліній даних, а

також може керувати індикаторами стану. У межах роботи передбачено, що мікроконтролер взаємодіє з програмним агентом через службовий канал, а вся логіка прийняття рішення залишається у програмній частині.

Для індикації стану можуть використовуватися світлодіоди, звуковий сигнал або невеликий дисплей. Зелена індикація позначає дозволений стан, червона — блокування або інцидент безпеки. Така індикація є корисною для користувача, оскільки дозволяє швидко зрозуміти, чи дозволено роботу пристрою. Водночас основним джерелом детальної інформації залишається програмний агент моніторингу.

Під час реалізації апаратної частини необхідно враховувати вимоги до розміщення компонентів. Елементи ESD-захисту мають бути розташовані біля роз'ємів, а лінії D+ і D- бажано вести поруч як диференційну пару. Компоненти, пов'язані з керуванням живленням, повинні мати достатні зазори та відповідати очікуваному струмовому навантаженню [потрібно уточнити].

Розроблена апаратна частина не є самостійним засобом прийняття рішень, але вона забезпечує важливий виконавчий рівень захисту. Її практичне значення полягає у можливості фізично припинити роботу невідомого або небезпечного USB-пристрою, що підвищує ефективність усього апаратно-програмного комплексу.

Під час реалізації апаратної частини потрібно забезпечити зрозумілу відповідність між командами програмного агента та фізичним станом фільтра. Наприклад, команда дозволу має призводити до замикання ліній даних і подачі живлення на зовнішній USB-пристрій, а команда блокування — до розмикання або вимкнення відповідних ліній. Така відповідність спрощує тестування і пояснення роботи комплексу.

Для макетної реалізації допускається використання модульного підходу, коли окремі вузли збираються з готових плат або доступних компонентів. Це дає змогу перевірити логіку роботи без виготовлення остаточної друкованої плати. Водночас у пояснювальній записці потрібно показати, як ці вузли можуть бути об'єднані у цілісний апаратний модуль.

У схемі важливо не змішувати лінії керування з лініями USB-даних. Команди від мікроконтролера мають керувати ключами та комутаторами, але не повинні безпосередньо впливати на диференційну пару USB без відповідних елементів. Такий підхід зменшує ризик нестабільної роботи та робить схему більш зрозумілою.

Апаратна частина також повинна мати визначений початковий стан. До моменту отримання команди від програмного агента доцільно тримати зовнішній USB-пристрій у заблокованому або обмеженому стані. Це запобігає ситуації, коли невідомий пристрій отримує доступ лише через те, що програмна частина ще не завершила перевірку.

Таблиця 3.1 – Функціональні вузли апаратної частини комплексу

Вузол	Призначення	Результат роботи
USB	ініціалізація USB-шини, enumeration, зчитування дескрипторів	первинний профіль пристрою
MCU	керування станами, попередня оцінка ризику, обмін з агентом	подія DEVICE_CONNECTED або ALERT
Gate	апаратне розмикання або дозвіл каналу живлення/даних	фізична протидія підозрілому пристрою
Storage	зберігання секрету HMAC, версії політик і службових лічильників	автентифікований модуль
UART	передавання службових повідомлень між MCU та агентом	узгоджене рішення системи

Таблиця 3.2 – Станова модель прошивки мікроконтролера

Стан	Умова переходу	Дія системи
1	2	3
INIT	подавання живлення або перезапуск	ініціалізація USB-host контролера і каналу зв'язку
WAIT	відсутній новий пристрій	очікування події підключення

Кінець таблиці 3.2

1	2	3
READ	виявлено пристрій	зчитування VID, PID, класів, інтерфейсів і серійного номера
EVAL	профіль сформовано	попередня оцінка ознак ризику та передавання події агенту
DECIDE	отримано рішення або виявлено критичну аномалію	відкриття або розмикання каналу підключення

Під час реалізації апаратної частини доцільно спочатку перевіряти базові стани керування без підключення потенційно небезпечного пристрою. Це дозволяє переконатися, що мікроконтролер коректно отримує команду, змінює стан виконавчих елементів і повертає відповідь програмному агенту. Лише після перевірки службових режимів доцільно переходити до тестів з реальними USB-пристроями. Така послідовність зменшує ризик помилкових висновків під час налагодження.

У практичній реалізації важливо, щоб апаратний модуль не вимагав складного обслуговування від користувача. Підключення повинно бути зрозумілим: один бік модуля з’єднується з робочою станцією, інший призначений для зовнішнього пристрою. Світлова або текстова індикація стану може спростити діагностику, але не повинна бути єдиним джерелом інформації. Основним джерелом для аналізу має залишатися журнал подій програмного агента.

3.2 Реалізація програмної частини комплексу

Інтерфейс не був перевантаженим і дозволяв продемонструвати ключову логіку комплексу.

Під час реалізації програмної частини враховано необхідність обробки помилок. Наприклад, зовнішній USB-пристрій може бути від’єднаний під час

перевірки, апаратний модуль може не відповідати, а параметри пристрою можуть бути неповними. У таких випадках система не повинна автоматично дозволяти роботу пристрою, а має зафіксувати подію та застосувати безпечну політику.

Реалізована програмна частина забезпечує зв’язок між користувачем, базою довірених пристроїв, політиками реагування та апаратним фільтром. Завдяки цьому комплекс може не лише повідомляти про підключення, а й виконувати практичну протидію невідомим або потенційно небезпечним USB-пристроям.

Реалізація графічного інтерфейсу має підтримувати основні ролі користувача. Звичайному користувачу достатньо бачити стан підключення та повідомлення про блокування. Адміністратору потрібен доступ до бази довірених пристроїв, політик реагування та журналу подій. Такий розподіл допомагає уникнути випадкової зміни критичних налаштувань.

У програмному агенті важливо передбачити зручне відображення причини блокування. Якщо пристрій заблоковано через невідомий клас, відсутність у базі або невідповідність параметрів, це має бути видно в інтерфейсі. Без такого пояснення користувач може сприймати систему як несправну, хоча вона фактично виконує політику безпеки.

Під час роботи з базою довірених пристроїв необхідно уникати автоматичного додавання будь-якого нового пристрою. Додавання має бути контрольованою дією, інакше база втратить сенс. У дипломній реалізації це може бути окрема кнопка або режим адміністратора [потрібно уточнити], але логіка повинна залишатися зрозумілою.

Обмін із мікроконтролером повинен бути достатньо простим і надійним. Команди можуть мати вигляд дозволу, блокування, запиту стану та підтвердження виконання. Якщо відповідь не отримано, агент не повинен вважати пристрій дозволеним. Така поведінка узгоджується з принципом безпечного стану.

Таблиця 3.3 – Структура програмних модулів системи

Модуль	Роль у системі	Основні дані
agent_core	запуск агента, вибір режиму, координація модулів	стан системи, активний профіль
hardware_adapter	обмін з MCU, перевірка автентичності модуля	READY, AUTH_RESPONSE, DEVICE_CONNECTED
policy_engine	застосування правил білого/чорного списку	VID, PID, serial, class, risk_score
device_inventory	зберігання довірених пристроїв та їх власників	ідентифікатор, призначення, строк дії
secure_logger	шифроване журналювання подій	тип події, час, рішення, HMAC
os_action_adapter	виконання блокування або дозволу на рівні ОС	device_id, команда реакції
cli_admin	обслуговування білого списку й перегляд журналу	команди адміністратора

Реалізація програмної частини передбачає обробку подій підключення у режимі, близькому до реального часу. Для цього програмний агент повинен періодично або подієво отримувати список активних USB-пристроїв і порівнювати його з попереднім станом. Якщо з’являється новий запис, запускається процедура перевірки. Такий механізм дозволяє відрізнити нове підключення від уже дозволеного пристрою та уникнути повторної обробки однієї й тієї самої події.

Для зручності супроводу логіку програми доцільно розділити на окремі функції або класи. Окремий компонент працює з конфігураційним файлом, окремий відповідає за перевірку пристрою, окремий формує команду для мікроконтролера, а окремий записує події до журналу. Такий підхід робить код зрозумілішим і полегшує тестування, оскільки кожен компонент можна перевіряти на окремих наборах вхідних даних.

3.3 Конфігурація та розгортання системи

Конфігурація системи передбачає підготовку програмного агента, апаратного USB-фільтра та початкових правил роботи. На першому етапі

необхідно визначити, які USB-пристрої вважаються довіреними для конкретного робочого місця. Це можуть бути службові накопичувачі, клавіатури, миші або інше обладнання, яке регулярно використовується користувачем і не повинно блокуватися під час звичайної роботи.

Після визначення довірених пристроїв їх параметри вносяться до бази довірених пристроїв. Для кожного запису доцільно зберігати не лише назву, а й технічні ознаки, отримані під час підключення. Це зменшує ризик помилкового дозволу у випадку підміни дескрипторів. Якщо пристрій змінює ключові параметри, система повинна вимагати повторної перевірки або застосовувати обмежувальну політику.

Наступним етапом є налаштування політик реагування. Адміністратор визначає, як система має діяти з невідомими знімними носіями, HID-пристроями, пристроями без серійного номера, пристроями з підозрілими дескрипторами або пристроями, що викликають перевантаження живлення. Політики можуть відрізнятися залежно від рівня захищеності робочого місця [потрібно уточнити].

Розгортання апаратної частини полягає у підключенні мікроконтролерного USB-фільтра між комп’ютером і зовнішнім USB-пристроєм. Такий модуль повинен отримувати живлення, мати службовий канал зв’язку з програмним агентом і забезпечувати комутацію ліній даних та живлення. Перед початком використання необхідно перевірити, що модуль коректно реагує на команди дозволу та блокування.

Програмний агент моніторингу встановлюється на робочу станцію та налаштовується для запуску разом із системою або вручну відповідно до умов використання [потрібно уточнити]. Він повинен мати доступ до інформації про USB-підключення, до бази довірених пристроїв і до каналу взаємодії з апаратним модулем. За потреби налаштовуються права доступу до журналу подій.

Під час початкового запуску доцільно виконати перевірку основних сценаріїв: підключення довіреного пристрою, підключення невідомого знімного носія, підключення HID-пристрою, від’єднання пристрою під час перевірки та втрата зв’язку з апаратним модулем. Такі дії дозволяють переконатися, що система

реагує передбачувано та не залишає небезпечний пристрій у дозволеному стані без рішення.

Окрему увагу слід приділити шифрованому журналюванню подій. Під час розгортання необхідно перевірити, що події створюються, зберігаються у захищеному вигляді та можуть бути переглянуті уповноваженою особою. Це важливо для подальшого аналізу інцидентів безпеки та підтвердження факту спрацювання політик реагування.

Конфігурація системи має бути достатньо простою, щоб її можна було виконати без складної інфраструктури. Саме це визначає практичну цінність комплексу для малого та середнього бізнесу, навчальних лабораторій і локальних мереж. Якщо рішення потребує надмірної кількості ручних дій, воно може бути незручним у щоденному використанні, тому налаштування мають бути зрозумілими та повторюваними.

У результаті розгортання система повинна переходити до робочого стану, в якому всі нові USB-підключення проходять контроль, рішення приймаються відповідно до політик, а події фіксуються у зашифрованому журналі. Такий підхід дозволяє організувати контроль підключень без повної відмови від використання USB-інтерфейсів.

Під час розгортання важливо визначити відповідальну особу, яка має право змінювати базу довірених пристроїв і політики реагування. Якщо ці дії будуть доступні кожному користувачу, рівень захисту знизиться. Тому навіть у простій реалізації потрібно розмежувати звичайне використання та адміністрування

Перед введенням системи в експлуатацію бажано провести початкове наповнення бази довірених пристроїв. Це дозволить уникнути ситуації, коли всі штатні пристрої одразу блокуються і заважають роботі користувача. Після такого наповнення система зможе відрізняти звичайні підключення від нових або підозрілих.

Конфігурація політик має відповідати умовам використання. У навчальній лабораторії може бути доцільним жорсткіше блокування знімних носіїв, тоді як у малому офісі може знадобитися контрольований дозвіл певних службових

накопичувачів. Тому система не повинна мати лише один незмінний режим роботи.

Після розгортання необхідно перевірити зручність щоденного використання. Якщо система створює надмірну кількість повідомлень або потребує постійного ручного підтвердження, користувачі можуть намагатися її обійти. Тому налаштування мають забезпечувати баланс між безпекою та працездатністю робочого процесу.

Конфігурація системи повинна починатися з визначення політики для типових робочих ситуацій. Наприклад, для навчальної лабораторії можна дозволити лише конкретні службові носії, заборонити невідомі HID-пристрої та переводити комбіновані пристрої в режим додаткової перевірки. Для офісного середовища політика може бути м’якшою, але вона все одно повинна фіксувати всі підключення. Наявність чітких правил полегшує пояснення роботи системи користувачам і адміністраторам.

Під час розгортання важливо перевірити не лише запуск програмного агента, а й його взаємодію з апаратним модулем після перезавантаження системи. Якщо служба стартує пізніше, ніж користувач підключає USB-пристрій, можливе коротке вікно ризику. Тому у конфігурації слід передбачити автоматичний запуск агента, контроль доступності мікроконтролера та початковий стан апаратного фільтра, який не надає невідомому пристрою повної довіри.

Під час підготовки до тестування необхідно визначити початкові умови, за яких результати можна вважати коректними. До таких умов належать однаковий стан бази довірених пристроїв перед кожним запуском, попередньо перевірений зв’язок із мікроконтролером, очищений або позначений журнал подій та використання однакового набору тестових пристроїв. Якщо ці умови не зафіксовані, складно порівнювати результати різних сценаріїв, оскільки відмінності можуть бути спричинені не логікою системи, а зміною середовища.

Окремо варто враховувати часовий показник реагування. Для USB-загроз важлива не лише правильність рішення, а й проміжок між появою пристрою та застосуванням політики. Якщо система виявляє підозрілий пристрій, але робить це із суттєвою затримкою, зловмисний код або емуляція введення можуть встигнути

виконати частину дій. Тому під час тестування доцільно фіксувати приблизний час реакції для дозволених, невідомих і заблокованих пристроїв.

Ще одним критерієм є зрозумілість повідомлень, які залишаються після тесту. Журнал подій повинен відображати не лише кінцевий результат, а й ключові параметри, що вплинули на рішення. Наприклад, для невідомого пристрою важливо бачити його клас, ідентифікатори, обрану політику та дію апаратного модуля. Завдяки цьому адміністратор може швидше визначити, чи було блокування обґрунтованим, і за потреби скоригувати правила.

3.4 Тестування апаратно-програмного комплексу

Тестування апаратно-програмного комплексу спрямоване на перевірку того, чи відповідає система поставленим задачам: контролює підключення USB-пристроїв, застосовує політики реагування, взаємодіє з мікроконтролерним USB-фільтром і зберігає події у захищеному вигляді. Оскільки робота має прикладний характер, тестування має відображати типові сценарії використання системи на робочому місці.

Першим сценарієм є підключення довіреного USB-пристрою. Система повинна отримати параметри пристрою, знайти відповідний запис у базі довірених пристроїв, застосувати політику дозволу та передати команду апаратному модулю для відкриття каналу. У журналі подій має бути зафіксовано факт підключення, параметри пристрою та прийняте рішення.

Другим сценарієм є підключення невідомого знімного носія. У цьому випадку пристрій не має відповідного запису в базі довірених пристроїв, тому система повинна застосувати політику для невідомих пристроїв. Залежно від налаштування це може бути блокування, очікування підтвердження або інша обмежувальна дія [потрібно уточнити]. Важливо, щоб пристрій не отримував повного доступу без перевірки.

Третім сценарієм є підключення невідомого HID-пристрою. Такий пристрій становить підвищений ризик через можливість HID-ін’єкції, тому система має реагувати жорсткіше, ніж на звичайний накопичувач. Очікуваною дією є блокування

або переведення пристрою в стан очікування до підтвердження. Подія повинна бути збережена в журналі як потенційний інцидент безпеки.

Окремо перевіряється реакція на підміну дескрипторів або часткову невідповідність параметрів. Якщо пристрій має знайому назву, але інші параметри відрізняються від запису в базі довірених пристроїв, система не повинна автоматично вважати його дозволеним. Такий сценарій демонструє важливість зіставлення сукупності ознак, а не лише одного текстового поля.

Для апаратної частини перевіряється виконання команд дозволу та блокування. Після команди дозволу зовнішній USB-пристрій має отримати можливість взаємодії з комп'ютером, а після команди блокування лінії даних або живлення мають бути розімкнуті. Якщо апаратний модуль не відповідає, програмний агент повинен зафіксувати подію та перейти до безпечного режиму.

Також перевіряється робота шифрованого журналювання подій. Після кожного сценарію в журналі має з'являтися відповідний запис із параметрами підключення, прийнятим рішенням і дією системи. Журнал потрібний не лише для відображення поточного стану, а й для подальшого аналізу інцидентів безпеки та контролю виконання політик.

У межах тестування не слід робити висновки про повний захист від усіх можливих USB-атак, оскільки така оцінка потребує окремих умов і розширеного набору перевірок. Метою тестування в цій роботі є підтвердження працездатності запропонованої архітектури, коректності основних сценаріїв і здатності системи виконувати практичну протидію невідомим або небезпечним підключенням.

Очікуваний результат тестування полягає в тому, що довірені пристрої допускаються до роботи, невідомі або небажані пристрої блокуються або переводяться в обмежений режим, а всі значущі події фіксуються у захищеному вигляді. Це підтверджує прикладну цінність розробленого комплексу для контролю USB-інтерфейсів.

Під час тестування важливо не обмежуватися одним успішним підключенням. Комплекс має бути перевірений у кількох режимах, оскільки саме різні сценарії

показують, чи правильно взаємодіють програмна та апаратна частини. Для цього доцільно перевіряти дозволені, невідомі, заборонені та помилкові підключення.

Окремо слід перевірити поведінку системи після перезапуску програмного агента або повторного підключення апаратного модуля. Стан системи не повинен ставати невизначеним. Якщо агент перезапускається, він має повторно отримати стан апаратного фільтра або перевести систему до безпечного режиму до завершення перевірки.

Тестування журналювання має включати не лише факт створення запису, а й перевірку його змісту. Запис повинен містити достатньо інформації для розуміння події: який пристрій було підключено, яку політику застосовано і яку дію виконано. Якщо журнал містить лише загальне повідомлення без параметрів пристрою, його практична користь зменшується.

Результати тестування доцільно описувати обережно, без надмірних тверджень. Можна стверджувати, що система реалізує контроль підключень і виконує передбачені дії в перевірених сценаріях. Водночас не слід заявляти про повний захист від усіх можливих USB-загроз, оскільки це потребувало б окремого розширеного випробування.

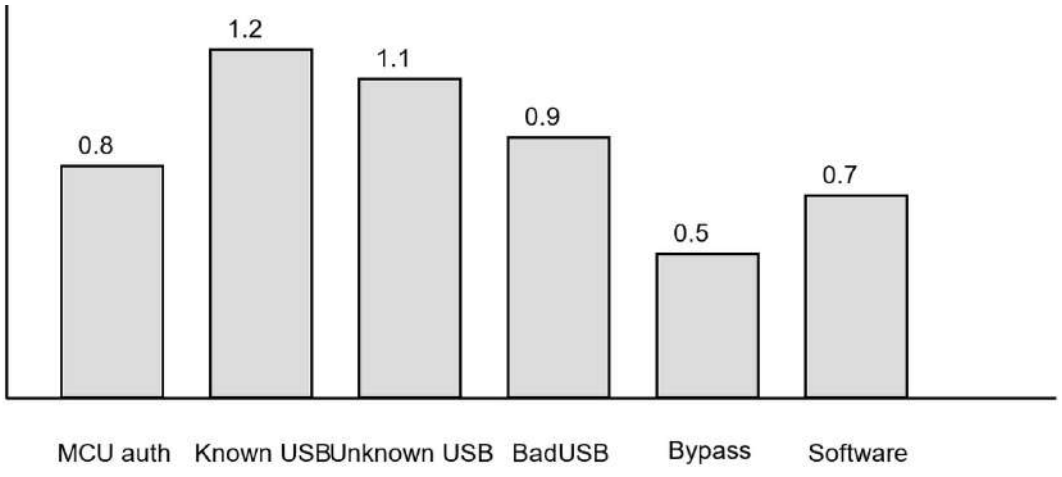


Рисунок 3.1 – Порівняння часу реакції системи

Таблиця 3.4 – Зведені результати тестування комплексу

Сценарій	Очікуваний результат	Фактичний результат	Час
Автентифікація MCU	Успішна перевірка	Успішно	0,8 с
Зареєстрований накопичувач	Дозволити	Дозволено	1,2 с
Незареєстрований накопичувач	Блокувати	Заблоковано	1,1 с
Емуляція BadUSB	Блокувати	Заблоковано	0,9 с
Обхід MCU	Блокувати	Заблоковано	0,5 с
Software-only	Робота без MCU	Коректно	0,7 с
Перевірка журналу	Розшифрування ключем	Успішнно	-

Тестування комплексу має охоплювати як штатні, так і небажані сценарії. До штатних належить підключення довіреного носія, повторне підключення вже відомого пристрою та коректне від’єднання після завершення роботи. До небажаних сценаріїв належать підключення невідомого накопичувача, імітація HID-пристрою, поява комбінованого пристрою та відсутність відповіді від апаратного модуля. Такий набір перевірок дозволяє оцінити систему не лише за успішними випадками, а й за поведінкою у проблемних умовах.

Результати тестування доцільно оцінювати за кількома критеріями: правильність прийнятого рішення, час реакції, повнота запису в журналі та відновлення системи після події. Якщо пристрій заблоковано, у журналі має бути зрозуміло, чому саме було прийнято таке рішення. Якщо пристрій дозволено, повинні бути зафіксовані параметри, на підставі яких він визнаний довіреним. Така деталізація робить результати тестування придатними для подальшого аналізу.

Тестові сценарії доцільно виконувати у такій послідовності, щоб кожен наступний сценарій перевіряв складнішу поведінку системи. Спочатку перевіряється дозволений пристрій, оскільки цей випадок підтверджує базову працездатність комплексу. Далі виконується підключення невідомого носія, що демонструє роботу політики обмеження. Після цього перевіряються сценарії з HID-

пристроєм, комбінованим пристроєм, відмовою апаратного модуля та переглядом журналу подій.

Під час інтерпретації результатів важливо не обмежуватися формальною ознакою успішності тесту. Якщо пристрій заблоковано, необхідно перевірити, чи не залишився він доступним для операційної системи у проміжному стані. Якщо пристрій дозволено, потрібно переконатися, що рішення прийнято саме через відповідність базі довірених пристроїв, а не через помилку перевірки. Такий підхід робить тестування ближчим до реальних умов експлуатації.

Повторюваність результатів є важливою ознакою коректної реалізації. Один і той самий пристрій за однакових умов повинен призводити до однакового рішення, а відмінності мають пояснюватися зміною конфігурації або правил. Якщо результат змінюється без очевидної причини, це може свідчити про нестабільну обробку подій, некоректну взаємодію з мікроконтролером або помилки у збереженні стану системи.

3.5 Висновки до розділу

У третьому розділі розглянуто реалізацію та перевірку апаратно-програмного комплексу для захисту USB-інтерфейсів. Основну увагу приділено практичній побудові системи, яка поєднує програмний агент моніторингу, базу довірених пристроїв, політики реагування, шифроване журналювання подій і мікроконтролерний USB-фільтр.

Реалізація апаратної частини передбачає використання керованого каналу між комп’ютером і зовнішнім USB-пристроєм. Мікроконтролерний USB-фільтр забезпечує можливість розмикання ліній даних і керування живленням, що дозволяє виконувати фізичну протидію невідомим або потенційно небезпечним пристроям.

Реалізація програмної частини забезпечує контроль підключень, аналіз параметрів USB-пристрою, перевірку за базою довірених пристроїв і застосування політик реагування. Програмний агент моніторингу також відповідає за взаємодію з

апаратною частиною та збереження подій у зашифрованому журналі.

Розглянуті сценарії тестування підтверджують логіку роботи комплексу: довірений пристрій може бути дозволений, невідомий USB-пристрій підлягає перевірці або блокуванню, а підозрілі підключення фіксуються як інциденти безпеки. Такий підхід дозволяє підвищити контроль за використанням USB-інтерфейсів на робочому місці.

Практичне значення реалізованого комплексу полягає у можливості застосування його як доступного засобу контролю USB-підключень у середовищах, де повна заборона USB-портів є незручною, але неконтрольоване використання зовнішніх USB-пристроїв створює ризики для інформаційної безпеки та цілісності обладнання.

Реалізація комплексу показує, що поєднання програмного агента та мікроконтролерного USB-фільтра є доцільним для задачі контролю USB-інтерфейсів. Програмна частина забезпечує аналіз і прийняття рішення, а апаратна частина виконує безпосередню дію щодо каналу підключення.

Запропонований підхід може бути використаний як основа для подальшого вдосконалення системи. Надалі можна уточнювати склад апаратних компонентів, розширювати політики реагування, удосконалювати графічний інтерфейс і додавати додаткові режими роботи [потрібно уточнити]. При цьому базова ідея контролю підключень і фізичного блокування залишається незмінною.

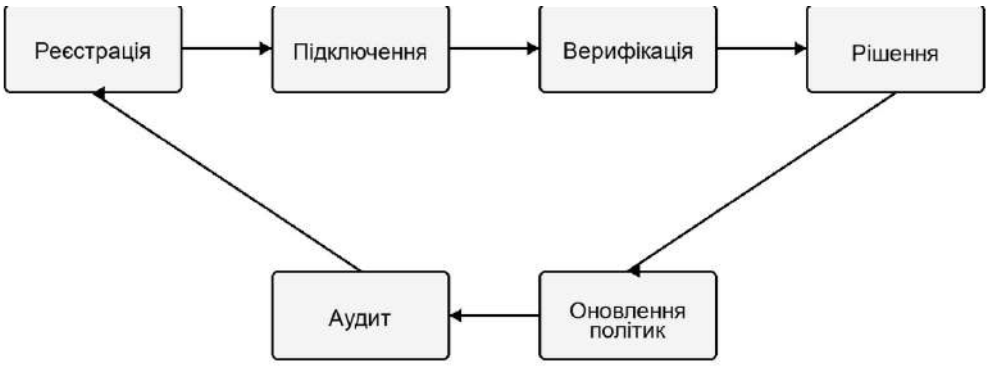


Рисунок 3.2 – Життєвий цикл USB-пристрою у системі контролю

Узагальнення результатів третього розділу показує, що практична цінність

комплексу полягає у поєднанні простоти реалізації з достатньою керованістю процесу реагування. Програмний агент забезпечує аналіз і журналювання, а апаратний модуль додає можливість фізичного обмеження каналу взаємодії. Завдяки цьому система не обмежується пасивним спостереженням за подіями, а може виконувати активну протидію у випадках, коли пристрій не відповідає політиці безпеки.

Подальше вдосконалення комплексу може бути пов’язане з розширенням набору правил, додаванням зручнішого інтерфейсу адміністратора, підтримкою централізованого зберігання журналів і порівнянням профілів пристроїв у часі. Водночас уже реалізований варіант демонструє базову працездатність запропонованого підходу та може використовуватися як навчальний або експериментальний прототип для дослідження захисту USB-інтерфейсів.

Додатково слід зазначити, що під час практичної реалізації особливе значення має узгодженість між програмною та апаратною частинами. Програмний агент може сформувати правильне рішення, однак без надійного виконання команди апаратним модулем захисна дія буде неповною. Аналогічно апаратний модуль не повинен самостійно надавати довіру пристрою без підтвердження з боку політики. Такий розподіл відповідальності робить комплекс більш контрольованим.

Практичний розділ також показує, що запропоноване рішення придатне для поступового розвитку. На початковому етапі достатньо реалізувати базові політики дозволу та блокування, після чого систему можна доповнювати детальнішими правилами, розширеним аналізом журналів і зручнішим інтерфейсом керування. Це важливо для навчального проєкту, оскільки дозволяє отримати працездатний результат без надмірного ускладнення архітектури.

Таким чином, реалізований комплекс підтверджує доцільність обраного підходу до захисту USB-інтерфейсів. Поєднання моніторингу, бази довірених пристроїв, журналювання та апаратного обмеження каналу забезпечує не лише виявлення підозрілих підключень, а й можливість своєчасної протидії. Це відповідає поставленій меті роботи та створює основу для подальшого

вдосконалення системи.

Під час оцінювання реалізації важливо враховувати, що запропонований комплекс орієнтований на локальне середовище, де адміністратор має змогу контролювати перелік дозволених USB-пристроїв. Саме тому база довірених пристроїв розглядається як центральний елемент політики безпеки. Вона не усуває потребу в контролі поведінки, але створює початковий фільтр, який відокремлює очікувані підключення від підозрілих.

Практичне використання такого підходу передбачає регулярне оновлення правил. Якщо в організації з’являються нові носії або периферійні пристрої, їх параметри потрібно вносити до бази після перевірки, а не дозволяти автоматично. Це дисциплінує процес експлуатації USB-портів і зменшує ймовірність випадкового підключення невідомого обладнання, яке може містити приховані функції.

Важливою перевагою розробленого рішення є можливість пояснити прийняте рішення через журнал подій. Для систем кібербезпеки це має суттєве значення, оскільки адміністратор повинен не лише побачити факт блокування, а й зрозуміти його причину. Наявність у журналі технічних ознак пристрою, часу події та застосованої політики підвищує прозорість роботи комплексу.

Разом з тим система не повинна розглядатися як єдиний засіб захисту від усіх можливих загроз. Вона доповнює антивірусний захист, політики операційної системи, резервне копіювання та організаційні правила. Її основна роль полягає у зменшенні ризику неконтрольованих USB-підключень і створенні додаткового бар’єра між зовнішнім пристроєм та робочою станцією.

Під час тестування підтверджується, що апаратно-програмний підхід є гнучкішим, ніж проста заборона USB-портів. Повна заборона може бути незручною для навчальних лабораторій або робочих місць, де знімні носії використовуються для легітимних задач. Запропонований комплекс дозволяє залишити можливість контрольованого використання пристроїв, але переводить невідомі підключення у керований сценарій перевірки.

Окремим напрямом подальшого розвитку може бути розширення механізму

аналізу поведінки після підключення. Наприклад, система може враховувати частоту появи нових інтерфейсів, спроби швидкого введення команд або зміну профілю пристрою протягом короткого проміжку часу. Такі ознаки можуть підвищити точність виявлення складніших атак, не змінюючи основної архітектури комплексу.

Також перспективним є централізоване зберігання журналів для кількох робочих станцій. У такому випадку адміністратор зможе бачити повторювані спроби підключення одного й того самого підозрілого пристрою на різних комп'ютерах. Це перетворює локальний засіб контролю на елемент ширшої системи моніторингу, що є корисним для навчальних закладів і невеликих організацій.

Отже, результати розділу підтверджують, що розроблений прототип виконує поставлені функції та демонструє практичну придатність запропонованого підходу. Система поєднує просту логіку розгортання, можливість фізичного блокування підозрілого пристрою та достатній рівень журналювання для аналізу інцидентів. Це створює підстави для формулювання загальних висновків за всією кваліфікаційною роботою.

З урахуванням отриманих результатів можна зазначити, що ефективність комплексу залежить від узгодженості трьох складових: правил довіри, коректної роботи програмного агента та надійного виконання апаратної команди. Якщо хоча б одна складова працює ізольовано, захист стає менш повним. Саме тому у роботі обрано апаратно-програмний підхід, у якому аналіз події та фізичне реагування доповнюють одне одного.

Важливим практичним аспектом є те, що система не потребує складної інфраструктури для базової роботи. Вона може бути розгорнута на окремій робочій станції, що робить її придатною для демонстраційного стенда, навчальної лабораторії або невеликого підрозділу. У такому середовищі особливо корисною є можливість швидко змінювати політику довіри та спостерігати наслідки цих змін під час тестування.

Розроблений підхід також дозволяє поступово підвищувати рівень захисту

без повної перебудови системи. Спочатку можна використовувати прості правила на основі ідентифікаторів і класу пристрою, а надалі додавати аналіз поведінки, контроль частоти підключень, розширене журналювання та централізоване збирання подій. Така масштабованість є важливою перевагою модульної архітектури.

Таким чином, підсумки практичного розділу підтверджують, що обрана архітектура є обґрунтованою для задачі виявлення та протидії загрозам через USB-інтерфейси. Система забезпечує контроль підключень, підтримує політики реагування, фіксує події безпеки та може виконувати блокування підозрілих пристроїв. Це дозволяє перейти до загальних висновків за результатами всієї роботи.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто проблему захисту комп'ютерних систем від загроз, що реалізуються через USB-інтерфейси, та розроблено апаратно-програмний комплекс для виявлення, запобігання і нейтралізації USB-атак. У межах роботи було поєднано теоретичний аналіз сучасних загроз, огляд наявних засобів захисту, побудову моделі загроз, проєктування архітектури комплексу, реалізацію його апаратної та програмної частин, а також перевірку працездатності запропонованого рішення.

Проведено комплексний аналіз сучасних загроз інформаційній безпеці, пов'язаних із використанням USB-інтерфейсів. Розглянуто основні типи USB-атак, зокрема BadUSB, HID-ін'єкції, атаки за допомогою пристроїв типу Rubber Ducky, шпигунські кабелі O.MG Cable, USB Killer, USB Drop Attack, викрадення даних через знімні носії та атаки з використанням мережевих USB-адаптерів. Визначено, що небезпека USB-атак полягає не лише у можливості передавання шкідливого коду, а й у здатності пристроїв маскуватися під легітимні периферійні засоби. Установлено, що загрози можуть стосуватися різних класів USB-пристроїв, зокрема Mass Storage, HID, CDC, Audio, Video та Vendor-Specific, що ускладнює їх своєчасне виявлення стандартними засобами операційної системи.

Проаналізовано існуючі засоби захисту від USB-загроз, серед яких апаратні рішення, програмні засоби контролю пристроїв та організаційні заходи безпеки. До апаратних засобів належать USB-блокатори, спеціалізовані фільтри та рішення типу USG. До програмних засобів належать Symantec Endpoint, McAfee Device Control, USBGuard, групові політики Windows та інші інструменти обмеження доступу до USB-портів. Порівняльний аналіз показав, що більшість таких рішень має обмеження: одні забезпечують лише фізичне блокування, інші працюють тільки на рівні операційної системи, а частина потребує значних фінансових витрат або складного адміністрування. Це підтвердило доцільність створення комбінованого апаратно-програмного комплексу.

Побудовано модель загроз для USB-інтерфейсів відповідно до методології STRIDE. У межах цієї моделі враховано можливість підміни пристрою,

несанкціонованого доступу, модифікації даних, приховування дій користувача або пристрою, витоку інформації та порушення доступності системи. На основі аналізу визначено основні принципи проєктування комплексу: нульова довіра до USB-пристроїв, ешелонований захист і мінімізація впливу людського фактора. Такий підхід дозволяє не покладатися лише на рішення користувача або стандартні механізми операційної системи, а застосовувати багаторівневу перевірку кожного підключення.

Спроєктовано апаратну частину комплексу, основою якої є USB-фільтр на базі Arduino Mega 2560 та USB Host Shield 2.0. Апаратний модуль виконує функцію проміжної ланки між зовнішнім USB-пристроєм і комп'ютерною системою. Його завданням є контроль підключення, аналіз параметрів пристрою, передавання інформації програмному агенту та виконання команд дозволу або блокування. Використання мікроконтролера дає змогу реалізувати додатковий рівень захисту, який не залежить повністю від операційної системи робочої станції.

Реалізовано програмну частину комплексу у вигляді агента на мові Python. Програмний агент має модульну архітектуру та відповідає за отримання даних про USB-пристрій, перевірку його ідентифікаторів, зіставлення з базою довірених пристроїв, застосування політик реагування і формування записів журналу подій. У роботі передбачено два режими функціонування комплексу: апаратний режим, у якому всі USB-пристрої проходять через захисний модуль, та програмний режим, у якому перевірка виконується засобами програмного агента за білим списком VID, PID і серійного номера.

Реалізовано механізм шифрованого журналювання подій безпеки. Журнал фіксує факт підключення пристрою, його основні параметри, результат перевірки, застосовану політику та прийняте системою рішення. Використання шифрування AES-256 у режимі CBC дозволяє підвищити конфіденційність і цілісність журналу інцидентів. Це важливо, оскільки журнал може містити службову інформацію про пристрої, політики доступу та спроби несанкціонованого підключення.

Проведено тестування апаратно-програмного комплексу у різних сценаріях роботи. Перевірено підключення довіреного пристрою, невідомого USB-носія, потенційно небезпечного HID-пристрою, а також сценарій, що імітує BadUSB-атаку.

Результати тестування показали, що система коректно визначає параметри пристроїв, приймає рішення відповідно до заданих політик і блокує несанкціоновані підключення. Час реакції комплексу у перевірених сценаріях становив приблизно 0,5-1,2 секунди, що є прийнятним для локального засобу контролю USB-підключень.

Розроблений комплекс є економічно доцільним рішенням, оскільки його апаратна частина може бути реалізована на доступних компонентах, а програмна частина не потребує придбання комерційних ліцензій. На відміну від багатьох готових рішень, запропонований комплекс поєднує програмну верифікацію, апаратне обмеження каналу підключення та захищене журналювання подій. Це робить його придатним для використання у навчальних лабораторіях, невеликих організаціях або як демонстраційний прототип системи контролю USB-інтерфейсів.

Таким чином, у д кваліфікаційній роботі досягнуто поставленої мети: розроблено апаратно-програмний комплекс для підвищення рівня захисту комп’ютерних систем від USB-загроз. Запропоноване рішення зменшує вплив людського фактора, оскільки підозрілі підключення можуть блокуватися автоматично, а всі значущі події фіксуються у захищеному журналі. Модульна архітектура комплексу створює можливості для подальшого розвитку, зокрема розширення правил верифікації, удосконалення інтерфейсу адміністратора, централізованого зберігання журналів та інтеграції з іншими засобами інформаційної безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Цирканюк Д., Соколов В. Методика розслідування інцидентів інформаційної безпеки. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2024. Т. 2, № 26. С. 140–154. DOI: 10.28925/2663-4023.2024.26.675

2. Колесник А. М., Ткач М. В. Пошук аномалій в системах автентифікацій за допомогою методів машинного навчання. XXI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (Київ, 11–12 травня 2023 р.): матеріали конференції. Київ: КПІ ім. Ігоря Сікорського, 2023. С. 387–391.

3. Gnatyuk S., Berdibayev R., Sydorenko V., Zhyharevych O., Smirnova T. Система корелювання подій та управління інцидентами кібербезпеки на об’єктах критичної інфраструктури. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2023. Т. 3, № 19. С. 176–196. DOI: 10.28925/2663-4023.2023.19.176196

4. Lakhno V., Blozva A., Husiev B., Osypova T., Matus Y. Інтегрування та захист IoT пристроїв у наявній інфраструктурі комп’ютерної мережі закладу освіти. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2021. Т. 3, № 11. С. 85–99. DOI: 10.28925/2663-4023.2021.11.8599

5. Що таке логи та як з ними працювати? URL: https://training.qatestlab.com/blog/technical-articles/logging-in-concepts-requirements-levels/ (дата звернення: 05.04.2025).

6. Що таке log file – Терміни та визначення у сфері кібербезпеки. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/log-file (дата звернення: 05.04.2025).

7. Григор’єв Д., Антипенко В. Актуальні тенденції оптимізації процесів логування та моніторингу в хмарних системах. Information Technology: Computer Science, Software Engineering and Cyber Security. 2024. № 2. С. 17–24. DOI: 10.32782/IT/2024-2-3

8. Pynbianglut Hadem, Saikia D. K., Moulik S. An SDN-based Intrusion Detection

System using SVM with Selective Logging for IP Traceback. Computer Networks. 2021. Vol. 191. DOI: 10.1016/j.comnet.2021.108015

9. Zhao Z., Xu C., Li B. A LSTM-Based Anomaly Detection Model for Log Analysis. Journal of Signal Processing Systems. 2021. Vol. 93. Pp. 745–751. DOI: 10.1007/s11265-021-01644-4

10. Abdulganiyu O. H., Ait Tchakoucht T., Saheed Y. K. A systematic literature review for network intrusion detection system (IDS). International Journal of Information Security. 2023. Vol. 22. Pp. 1125–1162. DOI: 10.1007/s10207-023-00682-2

11. Ahmad Z., Shahid Khan A., Wai Shiang C., Abdullah J., Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. 2021. Vol. 32. Article e4150. DOI: 10.1002/ett.4150

12. Qazi E.-u.-H., Imran M., Haider N., Shoaib M., Razzak I. An intelligent and efficient network intrusion detection system using deep learning. Computers and Electrical Engineering. 2022. Vol. 99. DOI: 10.1016/j.compeleceng.2022.107764

13. Alzahrani A. O., Alenazi M. J. F. Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. Future Internet. 2021. Vol. 13, No. 5. Article 111. DOI: 10.3390/fi13050111

14. Hidayat I., Ali M. Z., Arshad A. Machine Learning-Based Intrusion Detection System: An Experimental Comparison. Journal of Computational and Cognitive Engineering. 2022. Vol. 2, No. 2. Pp. 88–97.

15. Muhammad A. R., Sukarno P., Wardana A. A. Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. Procedia Computer Science. 2023. Vol. 217. Pp. 1406–1415. DOI: 10.1016/j.procs.2022.12.339

16. What Is An Intrusion Detection System (IDS)? URL: https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system (дата звернення: 11.03.2025).

17. Agoramoorthy M., Ali A., Sujatha D., F. M. R. T., Ramesh G. An Analysis of Signature-Based Components in Hybrid Intrusion Detection Systems. 2023 Intelligent

Computing and Control for Engineering and Business Systems (ICCEBS). Chennai, India. 2023. Pp. 1–5. DOI: 10.1109/ICCEBS58601.2023.10449209.

18. Rehman F., Mushtaq F., Zaman H. A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection for Enhanced Cybersecurity. 2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2). Islamabad, Pakistan. 2024. Pp. 1–7. DOI: 10.1109/ICoDT262145.2024.10740248.

19. Ahmed U., Nazir M., Sarwar A., et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports. 2025. Vol. 15. Article 1726. DOI: 10.1038/s41598-025-85866-7

20. Nguyen V. Q., Ngo T. L., Nguyen L. M., Nguyen V. H., Shone N. Deep Nested Clustering Auto-Encoder for Anomaly-Based Network Intrusion Detection. 2023 RIVF International Conference on Computing and Communication Technologies (RIVF). Hanoi, Vietnam. 2023. Pp. 289–294. DOI: 10.1109/RIVF60135.2023.10471853.

21. Acharya T., Annamalai A., Chouikha M. F. Efficacy of CNN-Bidirectional LSTM Hybrid Model for Network-Based Anomaly Detection. 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE). Penang, Malaysia. 2023. Pp. 348–353. DOI: 10.1109/ISCAIE57739.2023.10165088.

22. Ayad A. G., Sakr N. A., Hikal N. A. A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks. Journal of Supercomputing. 2024. Vol. 80. Pp. 26942–26984. DOI: 10.1007/s11227-024-06409-x

23. Karnan L., Mahalakshmi S. B., T. V. Hybrid Deep Learning Cloud Intrusion Detection. 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS). Krishnankoil, India. 2024. Pp. 1–6. DOI: 10.1109/INCOS59338.2024.10527729.

24. Satilmiş H., Akleylek S., Tok Z. Y. A Systematic Literature Review on Host-Based Intrusion Detection Systems. IEEE Access. 2024. Vol. 12. Pp. 27237–27266. DOI: 10.1109/ACCESS.2024.3367004.

25. Rastogi R., Yadav G., Sharma J., Singhwall J., Gupta M. Statistical Surveillance for Host-Based Intrusion Detection System (HIDS): An Intelligent System for Automation. In: Devi V. A. (eds) Sustainable IoT and Data Analytics Enabled

Machine Learning Techniques and Applications. Springer, Singapore. 2024. DOI: 10.1007/978-981-97-5365-9_5.

26. Du J., Yang K., Hu Y., Jiang L. NIDS-CNNLSTM: Network Intrusion Detection Classification Model Based on Deep Learning. IEEE Access. 2023. Vol. 11. Pp. 24808–24821. DOI: 10.1109/ACCESS.2023.3254915.

27. Nalini N., Chaudhary A., Surendran S., Muthuraja M., Ahmed I., N. T. J. Network Intrusion Detection System for Feature Extraction Based on Machine Learning Techniques. 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA). Coimbatore, India. 2023. Pp. 440–445. DOI: 10.1109/ICIRCA57980.2023.10220789.

28. Pillai S. E. V. S., Vallabhaneni R., Pareek P. K., Dontu S. Strengthening Cybersecurity using a Hybrid Classification Model with SCO Optimization for Enhanced Network Intrusion Detection System. 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT). Bengaluru, India. 2024. Pp. 1–9. DOI: 10.1109/ICDCOT61034.2024.10516247.

29. Li W., Varakantham P. Unsupervised Training Sequence Design: Efficient and Generalizable Agent Training. Proceedings of the AAAI Conference on Artificial Intelligence. 2024. Vol. 38, No. 12. Pp. 13637–13645. DOI: 10.1609/aaai.v38i12.29268

30. Liang J., Zhang S., Zhao R., Wu Y., Liu Y., Pan S. Omni-Frequency Channel-Selection Representations for Unsupervised Anomaly Detection. IEEE Transactions on Image Processing. 2023. Vol. 32. Pp. 4327–4340. DOI: 10.1109/TIP.2023.3293772.

31. Zhang T., Qiu H., Castellano G., Rifai M., Chen C. S., Pianese F. System Log Parsing: A Survey. IEEE Transactions on Knowledge and Data Engineering. 2023. Vol. 35, No. 8. Pp. 8596–8614. DOI: 10.1109/TKDE.2022.3222417.

32. Ma J., Liu Y., Wan H., Sun G. Automatic Parsing and Utilization of System Log Features in Log Analysis: A Survey. Applied Sciences. 2023. Vol. 13, No. 8. Article 4930. DOI: 10.3390/app13084930.

33. Das S., Tariq A., Santos T., Kantareddy S. S., Banerjee I. Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential

Research. In: Colliot O. (eds) Machine Learning for Brain Disorders. Neuromethods. 2023. Vol. 197. Humana, New York. DOI:10.1007/978-1-0716-3195-9_4.

34. Mienye I. D., Swart T. G., Obaido G. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. Information. 2024. Vol. 15, No. 9. Article 517. DOI:10.3390/info15090517.

35. Saravanan V., Madiajagan M., Rafee S. M., et al. IoT-based blockchain intrusion detection using optimized recurrent neural network. Multimedia Tools and Applications. 2024. Vol. 83. Pp. 31505–31526. DOI: 10.1007/s11042-023-16662-6.

36. Malashin I., Tynchenko V., Gantimurov A., Nelyub V., Borodulin A. Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review. Polymers. 2024. Vol. 16, No. 18. Article 2607. DOI: 10.3390/polym16182607.

37. Zhou H., Kang L., Pan H., et al. An intrusion detection approach based on incremental long short-term memory. International Journal of Information Security. 2023. Vol. 22. Pp. 433–446. DOI: 10.1007/s10207-022-00632-4.

38. Bahlali A. R., Bachir A., Cheriet A. Malicious Encrypted Network Traffic Detection Using Deep Auto-Encoder with a Custom Reconstruction Loss. 2023 International Symposium on Networks, Computers and Communications (ISNCC). Doha, Qatar. 2023. Pp. 1–7. DOI: 10.1109/ISNCC58260.2023.10323710.

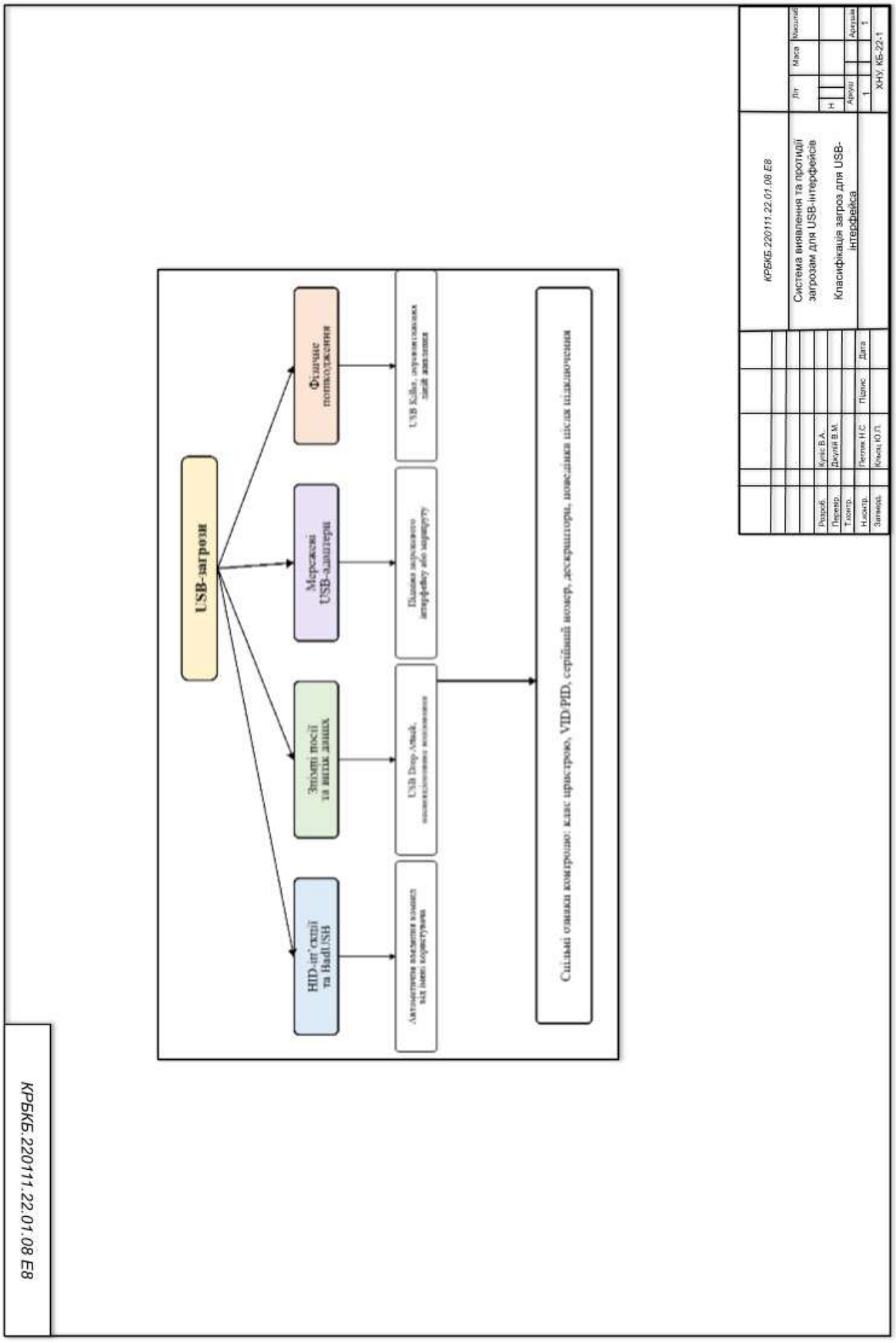
39. Kan D., Fang X. Event log anomaly detection method based on auto-encoder and control flow. Multimedia Systems. 2024. Vol. 30. Article 29. DOI: 10.1007/s00530-023-01199-3.

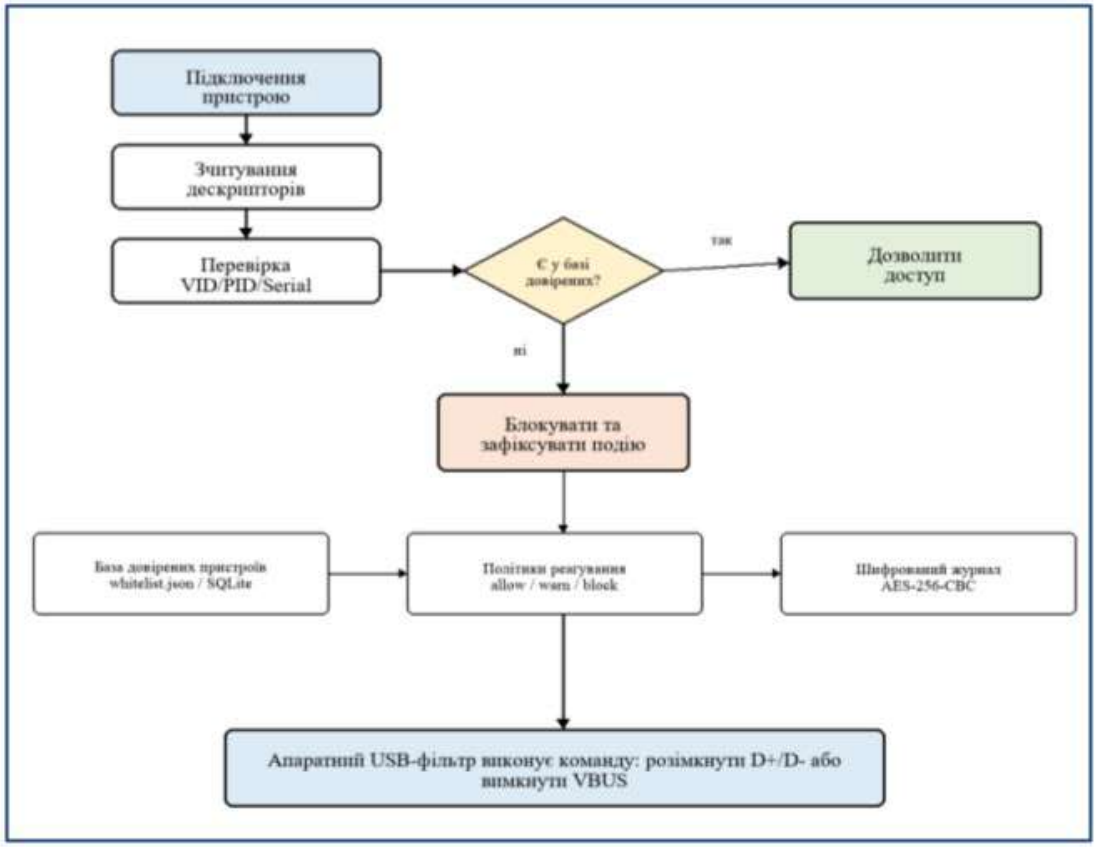
40. Li Z., Huang C., Qiu W. An intrusion detection method combining variational auto-encoder and generative adversarial networks. Computer Networks. 2024. Vol. 253. DOI: 10.1016/j.comnet.2024.110724.

41. СОУ 207.01:2017. Текстові документи. Загальні вимоги. Хмельницький: ХНУ, 2017. 46 с. URL:

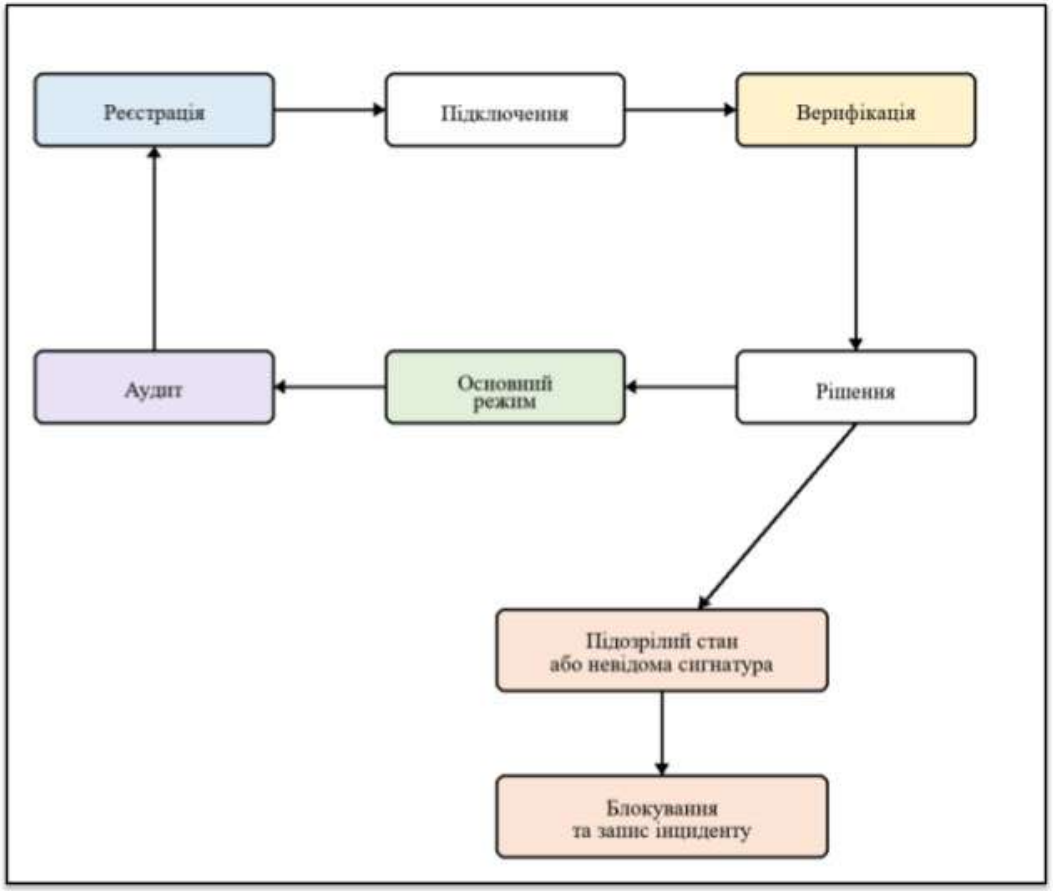
Додаток А

Копії графічної частини

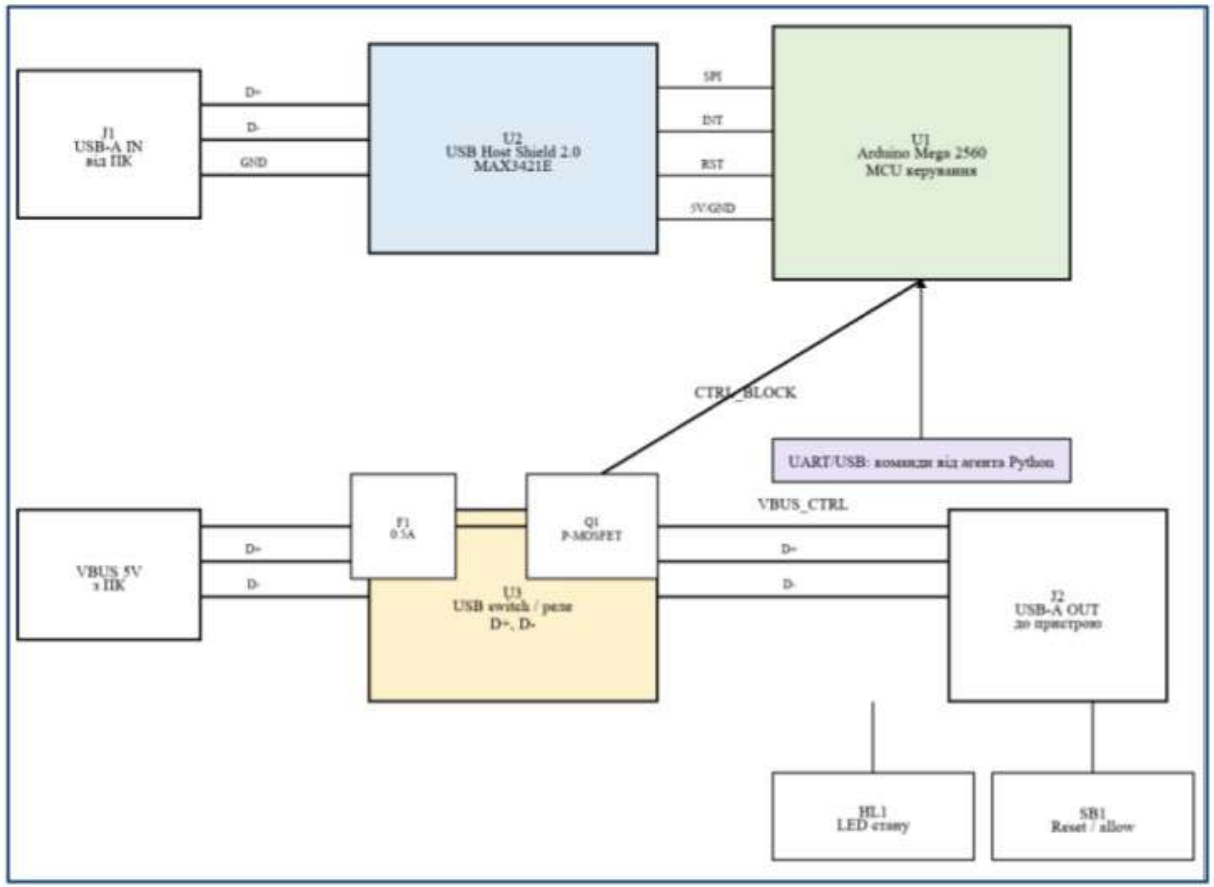




					КРБКБ.220111.22.01.08 Е8		
					Система виявлення та протидії загрозам для USB-інтерфейсів	Літ	Маса	Масштаб
Розроб.		Куліс В.А.			Алгоритм перевірки USB-пристрою	Н		
Перевір.		Джуган В.М.				Аркуш		Аркушів
Т.контр.						1		1
Н.контр.		Петлик Н.С.	Підпис	Дата		ХНУ, КБ-22-1		
Затверд.		Кльоц Ю.Л.						

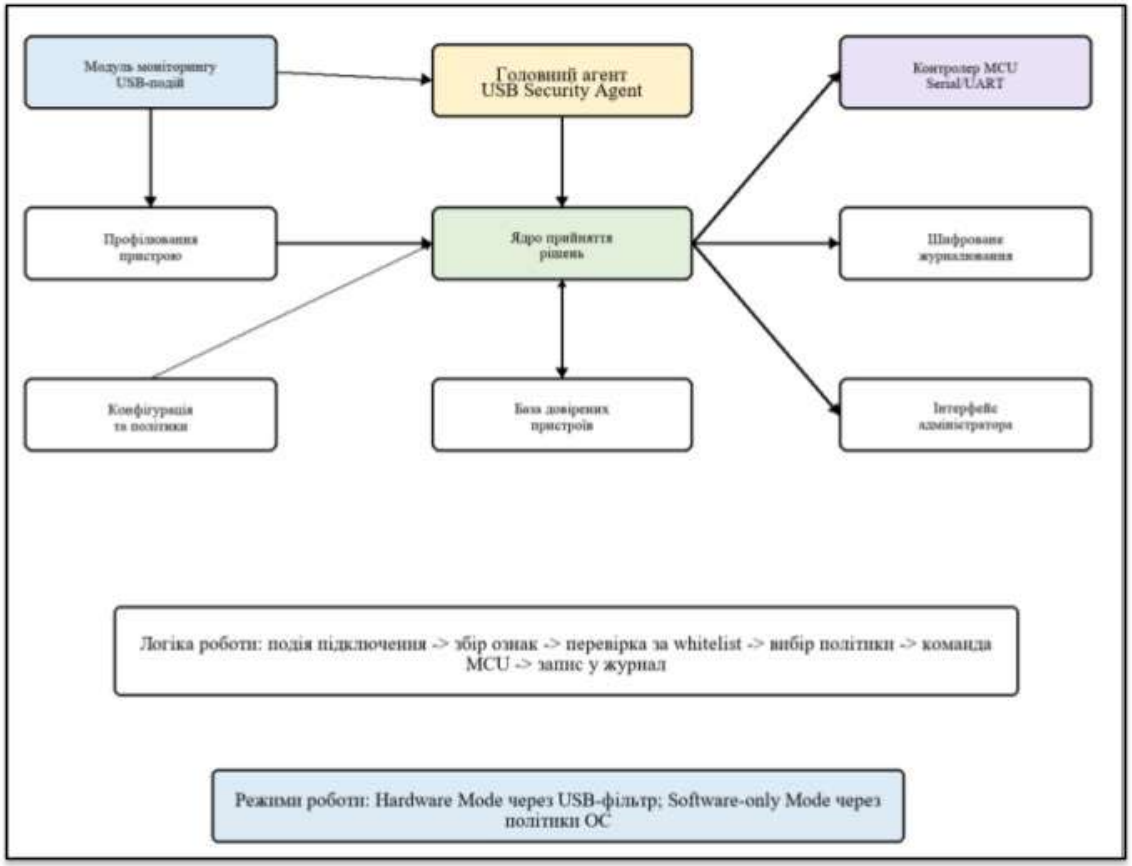


					КРБКБ.220111.22.01.08 Е8		
					Система виявлення та протидії загрозам для USB-інтерфейсів	Літ	Маса	Масштаб
Розроб.		Куліс В.А.			Життєвий цикл USB–пристрою у системі контролю	Н		
Перевір.		Джулій В.М.				Аркуш		Аркушів
Т.контр.						1		1
Н.контр.		Петлик Н.С.	Підпис	Дата		ХНУ, КБ-22-1		
Затверд.		Кльоц Ю.Л.						



					КРБКБ.220111.22.01.08 Е8		
					Система виявлення та протидії загрозам для USB-інтерфейсів	Літ	Маса	Масштаб
Розроб.		Куліс В.А.			Схема електрично принципова USB-інтерфейсів	Н		
Перевір.		Джулій В.М.				Аркуш		Аркушів
Т.контр.						1		1
Н.контр.		Петлик Н.С.	Підпис	Дата		ХНУ, КБ-22-1		
Затверд.		Кльоц Ю.Л.						

КРБКБ.220111.22.01.08 Е8



					КРБКБ.220111.22.01.08 Е8		
					Система виявлення та протидії загрозам для USB-інтерфейсів	Літ	Маса	Масштаб
Розроб.		Куліс В.А.			Блок-схема структура програмного забезпечення	Н		
Перевір.		Джулій В.М.				Аркуш		Аркушів
Т.контр.						1		1
Н.контр.		Петлик Н.С.	Підпис	Дата		ХНУ, КБ-22-1		
Затверд.		Кльоц Ю.Л.						