Міністерство освіти і науки України

«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій

Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В. о. завідувач кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«___» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 «Кібербезпека»
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: «Механізм захисту від атак типу BadUSB»

Виконавець: студент IV курсу, групи КБ-41

_____ **Дмитро ШЕРЕМЕТА**
(підпис) (ім'я та прізвище)

	Ім'я, прізвище	**Підпис**
Керівник	Леся БАРАНОВСЬКА	

Нормоконтроль	Лариса МИРУТЕНКО	

Київ 2025

Міністерство освіти і науки України

«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій

Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В. о. завідувач кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої-професійної програми)

Студенту **КБ-41** (група) **Шереметі Дмитру Олексійовичу** (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Механізм захисту від атак типу BadUSB

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Стандарти USB-протоколів, мікроконтролери, програмні засоби для розробки прошивок мікроконтролерів, атаки типу BadUSB, засоби виявлення та блокування несанкціонованих USB-пристроїв

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Архітектура USB-пристроїв та їх вразливості, принципи функціонування мікроконтролерів, механізм створення пристроїв типу BadUSB, класифікація та аналіз видів атак типу BadUSB, розробка та тестування

скетчів для Arduino Micro, реалізація HID-емуляції та атак типу fork-бомба, засоби захисту від BadUSB-атак, методи контролю USB-портів, технічні та організаційні рекомендації щодо запобігання несанкціонованому використанню USB-пристроїв

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Формування практичних рекомендацій для для забезпечення комплексного захисту від атак BadUSB

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видала _____ (підпис) Леся БАРАНОВСЬКА (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Дмитро ШЕРЕМЕТА (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 10.12.2024	*виконано*
2	Аналіз літератури	11.12.2024 – 29.12.2024	*виконано*
3	Дослідження архітектури та існуючих платформ створення BadUSB	30.12.2024 – 25.01.2025	*виконано*
4	Аналіз можливих атак BadUSB	26.01.2025 – 15.02.2025	*виконано*
5	Реалізація атаки BadUSB	16.02.2025 – 14.03.2025	*виконано*
6	Дослідження заходів та засобів захисту від атак BadUSB	15.03.2025 – 30.03.2025	*виконано*
7	Реалізація механізму захисту від атак типу BadUSB	31.03.2025 – 20.04.2025	*виконано*

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
8	Розробка рекомендацій для забезпечення комплексного захисту від атак типу BadUSB	21.04.2025 – 09.05.2025	*виконано*
9	Оформлення пояснювальної записки	10.05.2025 – 24.05.2025	*виконано*
10	Підготовка до захисту кваліфікаційної роботи	25.05.2025– 13.06.2025	*виконано*

Завдання видала _____ (підпис) Леся БАРАНОВСЬКА (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Дмитро ШЕРЕМЕТА (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, має 71 сторінку основного тексту, 2 таблиці та 43 рисунки. Список використаних джерел містить 21 найменування і займає 3 сторінки.

Мета роботи полягає у реалізації протидії атакам BadUSB, а також у створенні рекомендацій щодо захисту інформаційних систем від подібних загроз.

Для досягнення мети необхідно вирішити наступні *завдання*:

1. проаналізувати вразливості USB-пристроїв, архітектуру та методи створення BadUSB;
2. дослідити види та способи атак типу BadUSB;
3. реалізувати прототип атаки BadUSB на базі Arduino Micro;
4. реалізувати механізм захисту від подібних атак з використанням ПЗ Netwrix Endpoint Protector.

Об'єктом дослідження є процес захисту від атак типу BadUSB.

Предметом дослідження є механізм захисту від атак типу BadUSB.

Методи дослідження включають:

- аналіз літературних джерел;
- аналіз сучасних тенденцій;
- реалізацію практичних моделей;
- порівняльний аналіз.

Практична цінність полягає в запропонованих практичних рекомендаціях для забезпечення комплексного захисту від атак BadUSB.

Ключові слова: BadUSB, мікроконтролер, Arduino Micro, USB Rubber Ducky, емуляція пристроїв, DuckyScript, fork-бомба, кібербезпека, захист від USB-атак.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ СТВОРЕННЯ BADUSB 10
1.1 Мікроконтролери та вразливості USB-пристроїв 10
1.2 Перетворення USB-пристроїв на BadUSB 12
1.3 BadUSB на основі USB Rubber Ducky 14
1.4 BadUSB на основі плат Arduino 16
Висновки за розділом 1 19
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ВИДІВ ТА СПОСОБІВ АТАК ТИПУ BADUSB 21
2.1 Можливості BadUSB 21
2.2 Класифікація атак BadUSB 22
2.3 Розгляд атак BadUSB 26
2.4 Особливості BadUSB з Arduino Micro 28
2.5 Реалізація атаки BadUSB 30
Висновки за розділом 2 43
РОЗДІЛ 3 РЕАЛІЗАЦІЯ МЕХАНІЗМУ ЗАХИСТУ ВІД АТАК ТИПУ BADUSB 45
3.1 Розгляд механізмів захисту 45
3.2 Netwrix Endpoint Protector 47
3.3 Реалізація захисту від атаки BadUSB 48
3.4 Рекомендації щодо захисту від атак типу BadUSB 63
Висновки за розділом 3 64
ВИСНОВКИ 66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 69
ДОДАТОК 72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ВМ	–	Віртуальна машина
ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
ПК	–	Персональний комп’ютер
USB	–	Universal Serial Bus – універсальна послідовна шина
MSC	–	Mass Storage Class – клас пристроїв масового зберігання
HID	–	Human Interface Device – пристрій взаємодії з людиною
COM-port	–	Communications port – послідовний порт
IDE	–	Integrated Development Environment – інтегроване середовище розробки
Wi-Fi	–	Wireless Fidelity – технологія бездротового зв’язку
DNS	–	Domain Name System – система доменних імен
BIOS	–	Basic Input/Output System – базова система введення/виведення
UEFI	–	Unified Extensible Firmware Interface – уніфікований розширюваний інтерфейс мікропрограми
RAM	–	Random Access Memory – оперативна пам’ять
EEPROM	–	Electrically Erasable Programmable Read-Only Memory – електрично стирана програмована постійна пам’ять
IP	–	Internet Protocol – інтернет протокол

ВСТУП

Поява концепції BadUSB ознаменувала новий етап у розвитку атак на рівні апаратного забезпечення, коли звичайні пристрої, зокрема USB-флешки або клавіатури, перетворюються на повноцінні інструменти несанкціонованого доступу. Незважаючи на значний розвиток засобів антивірусного захисту та систем виявлення вторгнень, атаки типу BadUSB залишаються майже непомітними через використання легітимних каналів взаємодії на рівні USB-протоколу. Аналіз науково-технічної літератури та практичного досвіду фахівців у галузі інформаційної безпеки свідчить про зростаючу актуальність досліджень у сфері виявлення, протидії та моделювання таких загроз.

Актуальність роботи зумовлена стрімким поширенням USB-пристроїв у всіх сферах життєдіяльності, зокрема в Україні, де загрози інформаційній безпеці набувають особливої ваги в умовах кібервійни. Вразливості на рівні мікроконтролерів та USB-протоколів ускладнюють забезпечення захисту інформації, тому необхідність розробки дієвих методик тестування, виявлення і блокування BadUSB-пристроїв є вкрай нагальною як у науковому, так і в прикладному аспекті.

Робота має прикладну цінність для спеціалістів у сфері кібербезпеки та системного адміністрування, яка полягає в запропонованих практичних рекомендаціях для забезпечення комплексного захисту від атак BadUSB. Також робота може використовуватись як методичний матеріал для навчання безпечному поводженню з USB-пристроями.

Структура роботи включає три основні розділи.

Перший розділ присвячений аналізу принципів створення BadUSB-пристроїв. У ньому розглянуто архітектуру USB, особливості мікроконтролерів, здатних виконувати HID-емуляцію, а також описано найбільш поширені апаратні платформи для реалізації атак типу BadUSB, зокрема USB Rubber Ducky

та MalDuino на базі Arduino. Окрема увага приділяється мові DuckyScript та середовищам програмування.

Другий розділ зосереджений на класифікації та характеристиці можливих атак BadUSB. Детально описано типи атак – емуляційні, завантажувальні та комбіновані – їхні механізми, цілі та приклади реалізації. Також у цьому розділі розглянуто практичну реалізацію HID-атаки за допомогою Arduino Micro, зокрема запуск шкідливих команд через PowerShell та реалізацію fork-бомби у віртуальному середовищі.

Третій розділ присвячений методам захисту від BadUSB-атак. У ньому проаналізовано технічні та організаційні засоби протидії, описано інструменти контролю USB-портів, зокрема програму Netwrix Endpoint Protector. Подано приклад реалізації політик доступу до портів, а також результати тестування дієвості таких заходів у лабораторному середовищі. Було сформовано практичні рекомендації, щодо побудови надійної системи захисту від атак BadUSB з урахуванням сучасних засобів.

Таким чином, дана кваліфікаційна робота є цілісним дослідженням проблеми атак BadUSB, яке поєднує теоретичний аналіз, практичну реалізацію атаки та тестування захисних механізмів, і пропонує дієві заходи для підвищення кіберзахисту.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ СТВОРЕННЯ BADUSB

1.1 Мікроконтролери та вразливості USB-пристроїв

Universal Serial Bus (USB) – це стандарт роз’ємів і кабелів, який забезпечує передавання даних і живлення пристроїв, а також слугує універсальним інтерфейсом для послідовного обміну інформацією між пристроями через єдину шину. USB працює як послідовний порт – двонапрямний інтерфейс, що використовує послідовний-серійний зв’язок, тобто спосіб обміну даними, за якого інформація передається по одному біту за раз [1]. Це відрізняється від паралельного зв’язку, де всі біти передаються одночасно. Однією з основних цілей створення USB було розроблення ефективних протоколів обміну даними.

Однак ці протоколи не передбачають процедур перевірки автентичності та цілісності підключених USB-пристроїв. У поєднанні з відсутністю захисту мікроконтролерів від перепрограмування це створює потенційні ризики, які можуть бути використані зловмисниками для перетворення звичайних USB-пристроїв у небезпечні BadUSB [2].

BadUSB – це тип хакерських атак, які експлуатують вразливості USB-пристроїв. У таких атаках звичайний USB-пристрій або модифікується, або лише маскується під безпечний, щоб після підключення до цільового комп’ютера автоматично виконати шкідливий код [3; 4].

Кожен USB-пристрій має контролер, який забезпечує обмін даними з хостом – комп'ютером або іншим пристроєм, що надає ресурси та керує підключеннями. У простих USB-пристроях використовуються контролери з обмеженим набором функцій, які перетворюють команди центрального процесора в сигнали, зрозумілі пристрою, і навпаки. Їхня функціональність обмежена через відсутність різних функціональних блоків. Якщо для виконання

певної функції потрібних блоків немає, програмне забезпечення не може це компенсувати.

Функціональний блок – це набір мікроелектронних схем і компонентів, що виконують конкретну функцію, наприклад, постійна пам'ять для збереження даних чи програми.

У більш функціональних USB-пристроях застосовуються мікроконтролери – однокристальні мікрокомп'ютери, які мають універсальний набір функціональних блоків. Мікроконтролери включають мікропроцесор, оперативну і постійну пам'ять для збереження коду і даних, порти вводу-виводу та спеціалізовані блоки. Завдяки цим компонентам мікроконтролери є універсальними і можуть бути перепрограмовані для виконання різноманітних завдань, що робить можливим створення BadUSB.

Беручи до уваги наведені особливості, існує суттєва різниця між контролерами та мікроконтролерами, що визначає їхнє призначення та сферу застосування. Зіставлення характеристик (таблиця 1.1) дозволяє оцінити переваги кожного типу пристрою для конкретного застосування.

Таблиця 1.1

Ключові відмінності контролера та мікроконтролера

Характеристика	Контролер	Мікроконтролер
Функція	Виконує одну конкретну задачу	Виконує різноманітні завдання (програмований)
Склад	Спеціалізована логіка	Процесор + пам’ять + периферія
Програмування	Як правило, непотрібне (вбудована прошивка)	Потребує написання програми
Гнучкість	Обмежена, вузькоспеціалізована	Гнучкий у використанні
Приклад	Контролер USB, мережевий контролер	AT32UC3B1256, ATmega32U4

Відсутність захисту мікроконтролерів від перепрограмування дозволяє зловмисникам модифікувати прошивку пристрою – програму, вбудовану в апаратний пристрій для забезпечення його основних функцій, або замінювати її на шкідливу. Це дає змогу USB-пристрою імітувати інші типи обладнання та виконувати небажані дії, наприклад, вводити команди чи поширювати шкідливий код.

Наприклад, пристрій, який виглядає як звичайна USB-флешка, може виявитися USB-клавіатурою, яка швидко вводить команди від імені користувача. Більшість антивірусів не зреагують на таку атаку, адже вона виглядає як звичайний ввід із клавіатури.

Через різноманітність мікроконтролерів створення універсального шкідливого програмного забезпечення неможливе. Для кожного мікроконтролера потрібна окрема версія прошивки, а процес перепрограмування залежить від конкретної моделі. Це ускладнює масову підготовку пристроїв BadUSB, але не захищає від цілеспрямованих атак, особливо якщо зловмисник має доступ до пристрою-цілі.

Гнучкість і універсальність мікроконтролерів роблять BadUSB потужним інструментом для атак. Такий пристрій може збирати дані з хоста, аналізувати операційну систему і вибирати оптимальний вектор атаки. Все це робить пристрій BadUSB надзвичайно небезпечним.

1.2 Перетворення USB-пристроїв на BadUSB

Під час підключення мікроконтролер передає хосту інформацію про свій клас і функціональність. На основі цих даних хост завантажує відповідні драйвери для роботи з пристроєм. Один фізичний USB-пристрій може одночасно виконувати функції кількох класів. Наприклад, веб-камера може працювати як відеопристрій і як аудіопристрій.

Більшість атак типу BadUSB здійснюються через USB-флеш-накопичувачі – компактні пристрої для зберігання інформації, які використовують флеш-

пам'ять і підключаються до комп’ютера через USB-порт. Вони належать до USB Mass Storage Class (MSC), що є стандартом для USB-пристроїв, які відображаються на хост-комп’ютері як зовнішній жорсткий диск. Це дозволяє виконувати звичайні операції копіювання, видалення, перейменування та відкриття файлів. MSC використовує набір низькорівневих комунікаційних протоколів для взаємодії з хостом через USB-інтерфейс. Пристрої, які підтримують цей стандарт, називаються MSC-пристроями.

Щоб перетворити USB-флеш-накопичувач на BadUSB, його потрібно модифікувати так, щоб він міг емулювати інші USB-пристрої. Емуляція – це процес, при якому один пристрій або система імітує функціональність іншого пристрою чи системи, надаючи можливість виконувати їх функції без фізичної реалізації. Це дозволяє BadUSB працювати як:

– HID-пристрій (Human Interface Device), наприклад, клавіатура – вводить команди на комп’ютері без відома користувача;

– MSC-пристрій – емулює USB-флеш-накопичувач та одночасно виконує шкідливий код у фоновому режимі;

– USB-NIC (Network Interface Controller) – мережева карта, перенаправляє трафік через зловмисний сервер;

– USB-Serial – пристрої, які використовують інтерфейс USB для підключення до комп’ютера, але емулюють класичний серійний порт для передачі даних.

Більшість BadUSB емулюють клавіатуру, працюючи як Human Interface Device (HID) [5]. HID – це стандарт для підключення пристроїв введення/виведення до комп’ютера через USB-інтерфейс. Пристрій вважається HID, якщо він використовує цей стандарт для взаємодії з хост-системою. Наприклад, клавіатури та миші працюють за HID-протоколом, що забезпечує обмін даними між пристроєм та хостом. HID-стандарт також дозволяє пристрою повідомляти комп’ютеру про свої функції, наприклад, кількість кнопок у миші чи типи сигналів, які підтримує джойстик.

До HID-пристроїв відносяться:

– пристрої введення: клавіатури, миші, трекпади, стилуси;

– пристрої виведення: дисплеї, принтери, аудіопристрої, проектори.

Для перетворення USB-флеш-накопичувача на BadUSB необхідна наявність мікроконтролера [6]. Хоча USB-флеш-накопичувач уже має USB-інтерфейс і USB-контролер, цього недостатньо для реалізації, наприклад, HID-функціональності. З пристроїв, які мають лише USB-контролер, створити BadUSB неможливо через обмеження апаратної частини.

Найпопулярнішими пристроями з мікроконтролерами, що підтримують HID і не тільки, є:

– Rubber Ducky – спеціалізований USB-пристрій, створений для тестування безпеки [7];

– Arduino Leonardo/Micro – готові плати з мікроконтролерами, що мають вбудовану підтримку HID [8; 9];

– Digispark – компактна плата з бюджетним мікроконтролером, що підтримує роботу як HID-пристрій [10].

1.3 BadUSB на основі USB Rubber Ducky

Hak5 – компанія, яка спеціалізується на кібербезпеці, тестуванні на проникнення (англ. Pentesting) та створенні інструментів для досліджень у сфері інформаційної безпеки. Вона здобула популярність завдяки своїм інноваційним продуктам, навчальним матеріалам та шоу, орієнтованим на хакерів, ентузіастів технологій та професіоналів у сфері кібербезпеки. Одним із найвідоміших продуктів компанії є USB Rubber Ducky – USB-пристрій, який емулює клавіатуру для виконання скриптів – сценаріїв, що є послідовністю інструкцій для автоматичного виконання певних завдань із високою швидкістю

USB Rubber Ducky використовують у тестуванні на проникнення – процесі моделювання кібернападів на системи чи мережі для виявлення вразливостей,

які потенційно можуть бути використані зловмисниками. Rubber Ducky здобув популярність у 2010 році після демонстрації на конференції Defense Condition (DEF CON). З того часу він став символом простих, але ефективних атак, які легко реалізувати навіть без глибоких технічних знань. Це робить його корисним інструментом у навчальних курсах із кібербезпеки.

На технічному рівні USB Rubber Ducky побудований на основі мікроконтролера AT32UC3B1256 від Microchip Technology. Завдяки цьому мікроконтролеру пристрій здатний працювати в режимі HID, емулюючи клавіатуру чи мишу. Пристрій автоматично виконує задані сценарії або команди при підключенні до комп'ютера. Наприклад, він може швидко відкривати командний рядок, запускати програми чи виконувати дії, які дозволяють отримати доступ до системи.

USB Rubber Ducky використовує власну мову програмування під назвою DuckyScript. Вона за своєю суттю нагадує інтерпретовану мову, характерними рисами якої є: по-перше, виконання коду без попередньої компіляції, по-друге, простий синтаксис, і по-третє, легка розширюваність. Втім, від інтерпретованої мови DuckyScript успадкувала лише простий синтаксис та форму подання коду у вигляді скриптів. Насправді ж сценарії DuckyScript перед виконанням компілюються у спеціальний формат, який зчитується пристроєм. Ця мова спеціально створена для автоматизації дій, які пристрій виконує після підключення до комп'ютера. DuckyScript дозволяє записувати сценарії, які визначають, які саме команди будуть "набиратися" через клавіатуру, коли пристрій підключений до системи. Сценарії DuckyScript написані у текстовому форматі, де кожна команда відповідає конкретні дії [11; 12]. Наприклад:

```
DELAY 500
GUI r
DELAY 200
STRING cmd
ENTER
DELAY 500
```

```
STRING dir
ENTER
```

У цьому прикладі командний рядок відкривається за допомогою поєднання клавіш Windows + R, що відповідає команді GUI r.

Далі вводиться команда cmd, що відповідає команді STRING cmd.

Після цього виконується команда dir для перегляду файлів у поточній директорії, що відповідає команді STRING dir.

Важливою деталлю є затримка між виконанням команд, наприклад, DELAY 500. Оскільки невідомо, до якого комп'ютера буде підключено BadUSB – «швидкого» чи «повільного», – необхідно додати затримку у виконанні сценарію, щоб дати комп'ютеру достатньо часу для обробки та виконання команд.

Сценарії, написані на DuckyScript, потрібно компілювати у формат бінарних файлів, зрозумілих мікроконтролеру. Для цього Hak5 надає офіційний інструмент під назвою Ducky Encoder. Компіляція перетворює код високого рівня у машинний код – формат, який зберігає інформацію у вигляді послідовностей байтів (комбінацій нулів і одиниць) та виконується пристроєм. Це робить процес налаштування Rubber Ducky простим і зручним для використання.

Завдяки своїй гнучкості, швидкодії та можливості виконувати складні сценарії, USB Rubber Ducky став потужним інструментом у кібербезпеці. Він демонструє, наскільки дієвими можуть бути атаки за допомогою пристроїв, які на перший погляд здаються звичайними USB-накопичувачами.

1.4 BadUSB на основі плат Arduino

Arduino – це відкрита апаратно-програмна платформа, що надає широкий спектр можливостей для створення різних електронних проєктів [13]. Її основні переваги – доступність апаратних компонентів, таких як мікроконтролери й

друковані плати, а також зручне програмне середовище для написання, компіляції та завантаження коду в пристрої. Мікроконтролери Arduino програмуються мовою Wiring – спрощеним варіантом C та C++, що полегшує взаємодію з апаратними пристроями завдяки вбудованим бібліотекам і функціям.

У інтегрованому середовищі розробки Arduino або Arduino IDE (англ. – Integrated Development Environment) створюється скетч – програма, написана для мікроконтролера Arduino. Скетч складається з коду, який визначає, як мікроконтролер взаємодіятиме з підключеними пристроями. Він складається з двох основних частин [14]:

1. setup() – функція, яка виконується один раз при старті роботи мікроконтролера. Тут зазвичай налаштовуються параметри, ініціалізуються бібліотеки, підключаються зовнішні компоненти (наприклад, датчики, двигуни або світлодіоди).

2. loop() – функція, що виконується нескінченно в циклі після виконання setup(). Вона відповідає за основну логіку програми і керує діями мікроконтролера в реальному часі.

Апаратна частина платформи базується на мікроконтролерах серії ATmega від Microchip Technology. Моделі плат, такі як Arduino Nano, Arduino Uno та інші, відрізняються функціоналом і підходять для різних завдань. Програмна частина забезпечує інтуїтивне середовище для розробки коду, що дозволяє швидко створювати, тестувати й завантажувати проєкти на плату.

Для створення проєкту BadUSB ідеально підходять моделі Arduino Leonardo та Arduino Micro. Обидві ці плати оснащені мікроконтролером ATmega32U4, який має інтегрований USB-контролер. Це дозволяє їм працювати не лише як стандартний Communications port (COM-порт) – послідовний порт за стандартом Recommended Standard 232 (RS-232), але й як HID-пристрій (наприклад, клавіатура або миша). Завдяки такій функціональності ці плати ідеально підходять для емуляції пристроїв введення, що є ключовими для реалізації пристроїв BadUSB.

Варто зазначити, що під час роботи плати як віртуальний COM-порт використовуються програмні інтерфейси стандарту Communication Device Class (CDC), які емулюють відповідний послідовний порт через апаратний інтерфейс USB. Найчастіше COM-порт застосовується для зв'язку мікроконтролерів із комп'ютером або іншими пристроями.

Головною різницею між платами Arduino Leonardo та Arduino Micro є те, що остання має компактніший розмір, що забезпечує легкість реалізації прихованих або портативних версій BadUSB. Пристрої BadUSB, побудовані на основі плат Arduino, називаються MalDuinos, і вони широко використовуються для тестування безпеки та демонстрації вразливостей систем.

Як уже зазначалося, існує кілька підходів до створення BadUSB, і кожен із них має свої переваги та обмеження. Порівнюючи технології створення BadUSB (таблиця 1.2), можна зрозуміти, який пристрій краще підходить для конкретних завдань. Якщо потрібне готове рішення з мінімальними вимогами до налаштування, USB Rubber Ducky стане оптимальним вибором. Водночас BadUSB на базі плат Arduino надають більше можливостей для розширення та модифікації, що робить їх привабливими для користувачів, які хочуть налаштовувати пристрій під власні потреби.

Таблиця 1.2

Порівняння технологій створення BadUSB

BadUSB	Плат-форма	Реаліза-ція	Мікроконтролер	Середовище програму-вання	Мова програму-вання
USB Rubber Ducky	Hak5	Готове рішення	AT32UC3B1256	Ducky Encoder	DuckyScript
MalDuino	Arduino	Користу-вацький проєкт	ATmega32U4	Arduino IDE	Wiring

Висновки за розділом 1

У цьому розділі було розглянуто теоретичні основи та технічні підходи до реалізації пристроїв типу BadUSB. Питання безпеки USB-пристроїв є актуальним у контексті зростання кіберзагроз, зокрема з боку пристроїв, що можуть імітувати легітимні функції (наприклад, клавіатуру чи мережевий адаптер), але виконують шкідливі дії. Аналіз сучасних досліджень у цій галузі показав недостатній рівень захищеності на рівні апаратного розпізнавання USB-пристроїв, що створює передумови для зловживань, пов’язаних із концепцією BadUSB.

У роботі було проаналізовано основні принципи функціонування пристроїв BadUSB, які базуються на здатності USB-контролерів змінювати свій клас пристрою (наприклад, HID або CDC) та автоматично виконувати попередньо запрограмовані дії після підключення до системи. Розглянуто два основні підходи до реалізації BadUSB: за допомогою готового пристрою USB Rubber Ducky від Hak5 та користувацького рішення MalDuino, створеного на базі мікроконтролерів Arduino Leonardo та Arduino Micro. Проведене порівняння дозволило виділити ключові відмінності: USB Rubber Ducky відзначається простотою використання та фіксованим функціоналом, тоді як MalDuino забезпечує гнучкість, розширюваність та можливість кастомізації завдяки відкритій архітектурі Arduino.

Методологія реалізації MalDuino включає використання Arduino IDE, програмування на мові Wiring та використання можливостей мікроконтролера ATmega32U4 з вбудованим USB-контролером. Це дозволяє створювати пристрої, які можуть емулятивно виступати в ролі клавіатури або миші та автоматично вводити заздалегідь задані команди. Практична цінність цього підходу полягає в тому, що MalDuino не лише дозволяє здійснювати демонстрації потенційних атак, але й використовується в тестуванні інформаційної безпеки та в навчальних цілях.

Достовірність отриманих результатів забезпечується детальним технічним аналізом особливостей реалізації обох платформ, а також практично підтвердженими сценаріями використання. Здобуті результати дозволяють рекомендувати використання MalDuino як дієвого інструменту для вивчення атак типу BadUSB, оцінки вразливостей систем введення та розробки заходів протидії подібним загрозам. Також доцільним є впровадження механізмів апаратної автентифікації USB-пристроїв та контроль за їх поведінкою на рівні операційної системи як частина політики інформаційної безпеки.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ ВИДІВ ТА СПОСОБІВ АТАК ТИПУ BADUSB

2.1 Можливості BadUSB

До BadUSB пристроїв відносять USB-носії, що містять шкідливе програмне забезпечення (ПЗ). У таких випадках застосовують соціальну інженерію – мистецтво маніпулювання людьми з метою отримання конфіденційної інформації, доступу до систем або спонукання до виконання певних дій.

У випадку BadUSB такою дією є спонукання людини відкрити виконуваний файл – файл, який містить машинний код та може бути виконаний безпосередньо операційною системою або через командний інтерпретатор, тим самим запустивши шкідливе ПЗ. Для цього файл може мати будь-яку назву, яка здаватиметься доречною в конкретній ситуації. Наприклад, підкинувши USB-флеш-накопичувач в офіс, файл на ньому можна назвати «Договір». Часто трапляється, що хтось бажає відформатувати накопичувач і використати його у власних цілях, попередньо переглянувши його вміст [15].

Будь-який пристрій, що використовує інтерфейс USB для підключення до ПК, може бути BadUSB. Адже всередині невідомого пристрою може бути вбудоване апаратне забезпечення для виконання шкідливого коду.

Наприклад, такими пристроями можуть бути навіть USB-кабелі живлення. Люди часто вважають небезпечними лише USB-флеш-накопичувачі, тому без зайвих сумнівів використовують інші USB-пристрої, хоча вони також можуть становити загрозу.

BadUSB є небезпечними не лише для комп'ютерів, а й для смартфонів. Деякі пристрої оснащені USB Type-C, що дозволяє їм взаємодіяти з портативними гаджетами, такими як телефони. Відповідно, сценарії атак на

Android відрізняються від атак на Windows, але не обмежуються лише цими ОС – під загрозою також Linux і macOS.

BadUSB на базі плат Arduino можуть бути модифіковані додатковими модулями – окремими, самодостатніми елементами, що розширюють функціональність пристрою. MalDuino можна оснастити модулями Wireless Fidelity (Wi-Fi) – технологією бездротової передачі даних, що дозволяє підключатися до Інтернету та локальних мереж без використання кабелів [16]. Це відкриває можливість дистанційного керування BadUSB, що дозволяє запускати сценарії атаки віддалено.

Такі пристрої контролюються через веб-інтерфейс за допомогою власного Wi-Fi-хот-споту (англ. hot spot – «гаряча точка»). Він створює точку доступу до власної бездротової мережі, що дозволяє смартфону або ноутбуку підключатися безпосередньо до BadUSB. Це дає змогу редагувати, запускати, завантажувати й видаляти скрипти дистанційно. Такі пристрої значно просунуті і розширюють поверхню атак – набір методів, які зловмисник може використати для проникнення в систему та викрадення даних.

2.2 Класифікація атак BadUSB

Перед розглядом видів атак варто зазначити, що деякі з них можуть впливати на найнижчі рівні функціонування комп’ютерної системи, зокрема на системну прошивку.

Basic Input/Output System (BIOS) – це низькорівнева прошивка, що відповідає за початкову ініціалізацію обладнання та передає керування операційній системі. Його сучасною альтернативою є Unified Extensible Firmware Interface (UEFI), яка має розширені можливості, покращену безпеку та підтримку нових технологій.

Саме ці компоненти іноді стають цілями особливо небезпечних атак BadUSB, які передбачають зміну або пошкодження системного рівня.

Атаки BadUSB становлять серйозну кіберзагрозу, оскільки використовують один із найпоширеніших інтерфейсів – USB-порт. Пристрої BadUSB чудово підходять для непомітного виконання шкідливих дій, що можуть призвести до компрометації комп'ютерних систем – тобто ситуації, коли конфіденційна або захищена інформація стає доступною неуповноваженим особам, чи коли втрачається контроль над ресурсами (наприклад, обліковими записами, пристроями або даними). Існує кілька різновидів таких атак, які відрізняються принципом дії та технічними характеристиками пристроїв. Основні види атак BadUSB поділяються на завантажувальні, емуляційні та комбіновані, які поєднують риси перших двох. Кожен із цих типів має свої особливості залежно від обсягу пам’яті пристрою та способу його взаємодії з комп’ютерною системою, що впливає на рівень загрози в різних сценаріях атак.

Види атак BadUSB:

1. Завантажувальні – характерною ознакою таких атак є використання пристроїв BadUSB з великим об'ємом пам'яті. Завдяки цьому вони здатні зберігати шкідливе програмне забезпечення, яке згодом буде встановлюватися на комп'ютер.

2. Емуляційні – відповідно до назви, ці пристрої BadUSB під час атаки емулюють інші пристрої USB. Для цього їм не потрібно багато пам'яті, оскільки для збереження навіть великого сценарію атаки потрібно значно менше пам'яті, ніж завантажувальним BadUSB для зберігання шкідливого ПЗ.

3. Комбіновані – такі атаки здійснюються пристроями BadUSB, які водночас мають достатньо багато пам'яті та можуть емулювати інші пристрої USB.

Шкідливе програмне забезпечення має багато різних видів, а саме:

1. Вірус – це тип шкідливого програмного забезпечення, основною метою якого є поширення та виконання шкідливих дій на заражених системах. Для того щоб вірус потрапив у систему, зазвичай потрібна участь користувача – запуск зараженого файлу або програми. Після активації вірус автоматично виконується, розмножуючись шляхом копіювання себе в інші файли чи системні ділянки. У

багатьох випадках він працює відкрито, не маскуючись під легітимні процеси, хоча деякі його варіанти можуть використовувати техніки маскування.

2. Троянський кінь – це шкідлива програма, яка маскується під легітимне або корисне програмне забезпечення з метою непомітного проникнення в систему. Його завдання – забезпечити приховану присутність у системі для подальшого виконання шкідливих дій, не викликаючи підозри в користувача. Трояни зазвичай встановлюються разом із піратським ПЗ або маскуючись під інші програми, запускаються автоматично під час завантаження системи, але не розмножуються самостійно. Однак вони можуть виступати як інструмент для завантаження інших типів шкідливого ПЗ.

3. Хробак – це саморозмножуване шкідливе програмне забезпечення, яке здатне автономно поширюватися через комп’ютерні мережі без участі користувача. Метою хробака є максимальне розповсюдження на інші пристрої з подальшим запуском шкідливих дій. Він використовує системні вразливості або слабкі паролі для проникнення та активується автоматично після зараження. Хробак може діяти як приховано, так і відкрито, іноді спричиняючи помітні навантаження на ресурси системи.

4. Шпигунське програмне забезпечення (англ. Spyware) призначене для прихованого збору конфіденційної інформації про користувача, такої як паролі, особисті дані чи історія перегляду. Воно часто встановлюється разом із неліцензійними програмами або проникає через веббраузери та інсталятори. Після потрапляння в систему spyware запускається автоматично, діє приховано та не має функцій самостійного поширення. Для маскування своєї активності таке ПЗ імітує звичайні процеси операційної системи.

5. Руткіт (англ. Rootkit)– це особливо небезпечний тип шкідливого ПЗ, що дозволяє зловмиснику отримати прихований і постійний доступ до системи. Його мета – забезпечити невидимість присутності шкідливих компонентів та контроль над системою, при цьому уникаючи виявлення захисними засобами. Руткіти можуть потрапляти в систему разом із піратським ПЗ або під виглядом корисних програм, запускаються автоматично й функціонують приховано. Вони

здатні змінювати системні файли, приховувати процеси та мережеву активність, ускладнюючи виявлення.

6. Бекдор (англ. Back door) – це не окрема програма, а вразливість або навмисно створений обхід захисних механізмів, що дозволяє зловмиснику отримати несанкціонований доступ до системи. Бекдор може бути реалізований як програмний інтерфейс, залишений розробником, або впроваджений через інше шкідливе ПЗ, наприклад, троян чи руткіт. Він забезпечує контроль над системою, дозволяючи обійти аутентифікацію чи інші засоби безпеки.

Загалом, вірус, троян, хробак і шпигунське ПЗ – це різні форми шкідливого програмного забезпечення, кожна з яких має свої механізми проникнення, поширення та дії. Бекдор, на відміну від них, не завжди є самостійною програмою, але часто використовується як канал для забезпечення віддаленого контролю над системою внаслідок дій зловмисника.

BadUSB-атаки реалізуються у різних формах, проте їх основною метою є підключення до комп’ютера-цілі. Після цього можуть реалізовуватись різні сценарії атаки. З огляду на це доцільно класифікувати такі атаки відповідно до їхньої кінцевої мети.

Класифікація атак BadUSB відповідно до мети:

– Отримання несанкціонованого доступу:
  - додавання нового адміністративного облікового запису;
  - зміна налаштувань безпеки системи;
– Крадіжка даних:
  - експортування збережених паролів;
  - витяг файлів із жорсткого диска або буфера обміну;
  - зчитування cookie-файлів або автозаповнення браузера;
  - завантаження та встановлення шпигунського пз;
– Компрометація мережі:
  - зміна мережевих налаштувань;
  - запуск сканування внутрішньої мережі;

- встановлення проксі-сервера для перехоплення трафіку;

– Віддалене управління:

- створення бекдора;
- встановлення троянів або руткітів;

– Фізичне знищення:

- видалення системних файлів;
- форматування жорсткого диску або інших накопичувачів;
- спалення контролера usb;
- пошкодження материнської плати;
- запис шкідливого коду в BIOS/UEFI;

– Саботаж:

- створення нескінченних процесів;
- відкриття тисячі вкладок у браузері для завантаження оперативної пам’яті;
- шифрування файлів із подальшим вимаганням викупу;
- завантаження та виконання вірусів, троянів, шкідливого ПЗ.

2.3 Розгляд атак BadUSB

Атаки типу BadUSB можуть мати різні форми, що залежать від способу взаємодії з комп'ютерною системою та використовуваних технологій. Кожна з цих атак має свої особливості та потенціал для завдання шкоди. Розглянуто найбільш поширені типи атак BadUSB, які включають емулювання клавіатури, атаку на мережеві підключення, а також більш складні методи, такі як зараження на етапі завантаження:

1. Емуляція клавіатури – при підключенні BadUSB пристрій повідомляє комп’ютеру жертви, що він є клавіатурою, а через певний час починає надсилати послідовності натискань клавіш. Це дозволяє зловмиснику виконувати будь-які дії, доступні авторизованому користувачу, використовуючи лише клавіатуру.

Наприклад, зловмисник може завантажити та запустити шкідливе програмне забезпечення з інтернету.

Основний недолік цього виду атак – відсутність доступу до інформації на екрані, що унеможливлює зворотний зв'язок. Однак у просунутих BadUSB з модулем Wi-Fi є можливість запускати різні сценарії віддалено, поки не буде досягнуто бажаного результату.

2. Емуляція мережевої карти – BadUSB підключається до комп'ютера і видає себе за мережеву карту (тобто пристрій, який забезпечує підключення до інтернету або локальної мережі). Далі BadUSB повідомляє персональному комп'ютеру (ПК), що потрібно використовувати новий DNS-сервер – систему, яка перетворює доменні імена в IP-адреси і навпаки. Але при цьому він не надає шлюз за замовчуванням, тобто не стає пристроєм, який направляє трафік на зовнішні мережі. У результаті DNS-сервером стає ПК зловмисника, але через відсутність шлюзу за замовчуванням для доступу в інтернет використовується справжній мережевий інтерфейс.

Таким чином, пристрій BadUSB є підґрунтям для атаки «Людина посередині» (MITM-атака, англ. Man in the Middle). У цій атаці зловмисник перехоплює і часто змінює комунікацію між двома сторонами, не будучи поміченим. Вона зазвичай відбувається, коли дві сторони (наприклад, користувач і сервер) взаємодіють між собою через мережу, і зловмисник «стає» між ними.

Під'єднавши BadUSB до свого ПК, користувач стає жертвою: атакуючий отримує контроль над його DNS-запитами та може перенаправляти їх на шкідливі сайти. Це може призвести до крадіжки даних, таких як логіни, паролі або фінансова інформація, через підроблені сторінки.

3. USB Kill-атака – дана атака використовує спеціально модифікований USB-пристрій (так званий USB Killer) для виведення з ладу електронного обладнання шляхом подачі високої напруги через USB-порт [17]. Такий пристрій виглядає як звичайна флешка, але всередині містить схему, що накопичує електричний заряд у конденсаторах і різко скидає його назад у лінії живлення

USB. У більшості випадків це призводить до перегорання контролерів живлення, пошкодження материнської плати та виходу пристрою з ладу.

4. Втеча з віртуального середовища – ця атака використовує можливість повторної ініціалізації пристрою. BadUSB пристрій під’єднаний до віртуальної машини, автоматично запускає вірус, який заражає пристрої підключені через USB. Після цього заражена мікропрограма ініціалізується повторно, створюючи два незалежні пристрої: нове, яке буде автоматично підключене до хостової ОС, і вже існуюче, що залишиться у віртуальному середовищі. У результаті зловмисник може вийти за межі віртуальної машини та отримати доступ до хостової операційної системи.

5. Атака на BIOS/UEFI – атака BadUSB на BIOS/UEFI є однією з найнебезпечніших загроз, оскільки вона здатна змінити прошивку материнської плати, що може зробити комп'ютер непридатним для використання навіть після перевстановлення операційної системи. Ця атака дозволяє зловмисникам записати шкідливий код безпосередньо у BIOS/UEFI, змінюючи його роботу або вбудовуючи бекдор.

6. Зараження на етапі завантаження (англ. Boot Injection) – BadUSB пристрій із достатнім обсягом пам’яті може виявити момент увімкнення комп’ютера та на етапі ініціалізації BIOS передати для завантаження вірус. Це можливо завдяки аналізу поведінки хост-машини під час взаємодії з USB-контролером, що дозволяє визначити тип ОС (Windows, Linux чи macOS), а також версію BIOS.

2.4 Особливості BadUSB з Arduino Micro

Розглянемо пристрій BadUSB, зібраний на основі плати Arduino Micro, що використовує ATmega32U4 – мікроконтролер від компанії Microchip Technology, який має вбудовану апаратну підтримку USB. Це означає, що плата може працювати в режимі емуляції клавіатури. Наприклад, вона здатна надсилати

сигнали до комп’ютера, імітуючи натискання клавіш. Подібні плати часто мають компактні розміри, що дозволяє зручно використовувати їх у різних проєктах.

Мікроконтролер ATmega32U4 має три види пам’яті [18]: програмну, оперативну та незалежну. Їхні інші назви відповідно – Flash-пам’ять, Random Access Memory (RAM) та Electrically Erasable Programmable Read-Only Memory (EEPROM).

Flash-пам’ять – це постійна пам’ять, у якій зберігається прошивка (програма, яку виконує мікроконтролер). Навіть після вимкнення живлення дані залишаються збереженими. У ATmega32U4 обсяг Flash-пам’яті становить 32 КБ, з яких 4 КБ зайняті завантажувачем (англ. Bootloader) – спеціальною програмою, яка виконується одразу після вмикання мікроконтролера і відповідає за завантаження основної прошивки.

RAM – це тимчасова пам’ять, у якій під час виконання програми зберігаються змінні, результати проміжних обчислень, стек викликів (структура даних типу Last In, First Out (LIFO), тобто «останній увійшов – перший вийшов»), а також буфер – область пам’яті, призначена для тимчасового зберігання даних під час їх передавання між пристроями або програмами. Після вимкнення живлення всі дані з RAM зникають. Обсяг цієї пам’яті в ATmega32U4 становить 2,5 КБ (тобто 2560 байт).

EEPROM – це пам’ять для збереження даних, які мають залишатися після вимкнення живлення, але які можна змінювати програмно. У ATmega32U4 її обсяг становить 1 КБ (1024 байти). Вона використовується для збереження параметрів, налаштувань або лічильників, які мають зберігатися між перезавантаженнями. Запис у EEPROM відбувається повільніше, ніж у RAM або Flash.

Варто зазначити, що EEPROM розрахована на 100 000 і більше циклів запису, а Flash – приблизно на 10 000. Якщо часто змінювати дані у Flash, вона швидко вийде з ладу. Попри це, USB-флеш-накопичувачі використовують Flash-пам’ять як незалежну пам’ять. Такі накопичувачі мають спеціальний контролер пам’яті, який:

– оптимізує процес запису (записує дані по секторах, мінімізуючи зношування);

– рівномірно розподіляє навантаження;

– підтримує файлові системи (наприклад, File Allocation Table (FAT32), New Technology File System (NTFS)), що дозволяють зручно зберігати дані.

Мікроконтролер ATmega32U4 не має такого контролера і не підтримує файлові системи, тому запис у Flash є складним і обмеженим.

Як зазначалося раніше, комбіновані атаки BadUSB здійснюються пристроями, які одночасно мають великий обсяг пам’яті та здатні емулювати різні типи USB-пристроїв. Такі пристрої можуть одночасно виконувати функції кількох класів. Наприклад, BadUSB-пристрій може діяти як USB-флешка і клавіатура водночас. У такому разі зловмисник має достатньо пам’яті для збереження шкідливого програмного забезпечення і може встановити його на комп’ютер-ціль за допомогою емуляції клавіатури. Цей підхід також дозволяє збирати великі обсяги даних і зберігати їх на самому BadUSB-пристрої.

2.5 Реалізація атаки BadUSB

Для програмування та безпечного тестування BadUSB-пристрою необхідно мати контрольоване середовище – а саме, віртуальну машину.

Віртуальна машина (ВМ) – це програмна емуляція комп’ютера, яка працює на фізичному комп’ютері (хості) та дозволяє запускати операційну систему (гостьову ОС) і програми так, ніби це окремий фізичний пристрій. Віртуальна машина має власні віртуальні апаратні ресурси: процесор, оперативну пам’ять, жорсткий диск, мережеву карту тощо, які фактично використовують ресурси хост-системи.

Віртуальні машини створюються й керуються за допомогою спеціального програмного забезпечення – гіпервізора (наприклад, Oracle VM VirtualBox, VMware).

Налаштовано віртуальну машину для роботи з пристроєм BadUSB.

Для цього до віртуальної машини додається фізичний USB-порт хоста. Машину вимикають, переходять до її налаштувань, відкривають вкладку USB, додають новий фільтр (див. рис. 2.1).

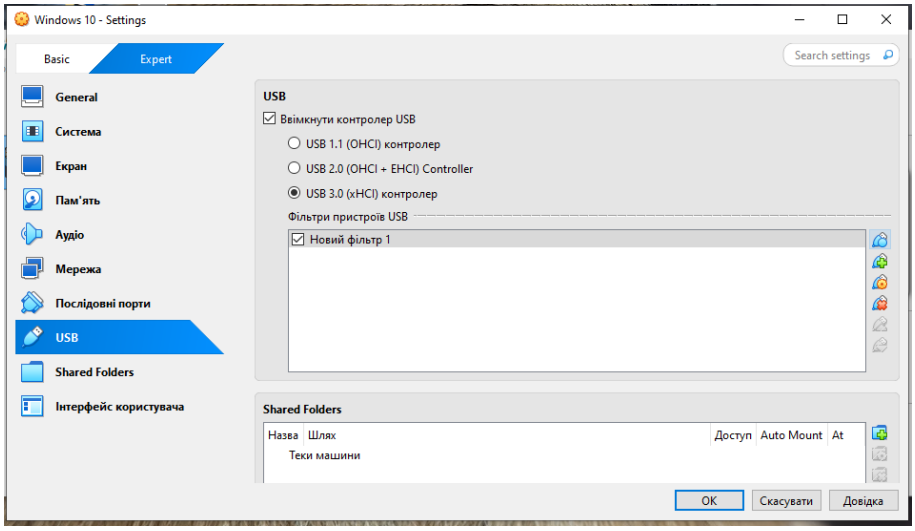

Рисунок 2.1 – Створення нового фільтру USB-пристроїв

Редагують фільтр (див. рис. 2.2): зазначають його назву та номер фізичного порту хоста.

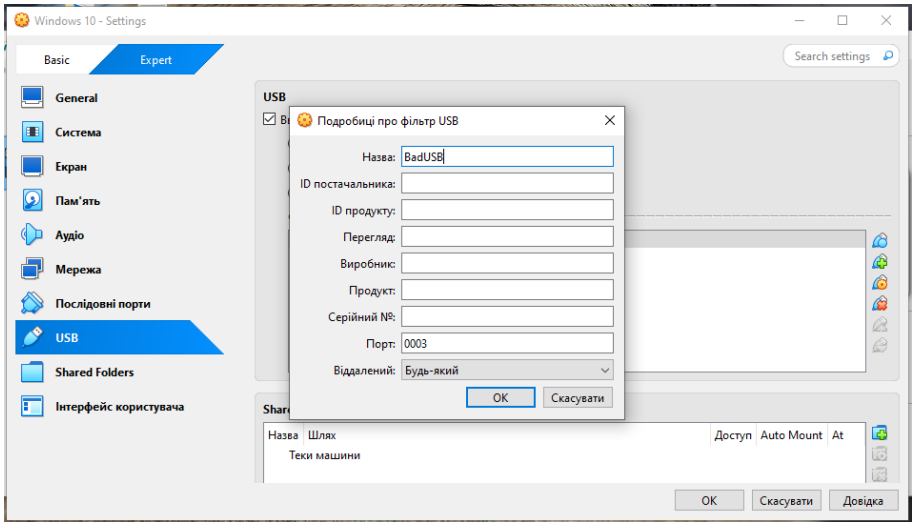

Рисунок 2.2 – Редагування створеного фільтру USB-пристроїв

Натискають «ОК» (див. рис. 2.3).

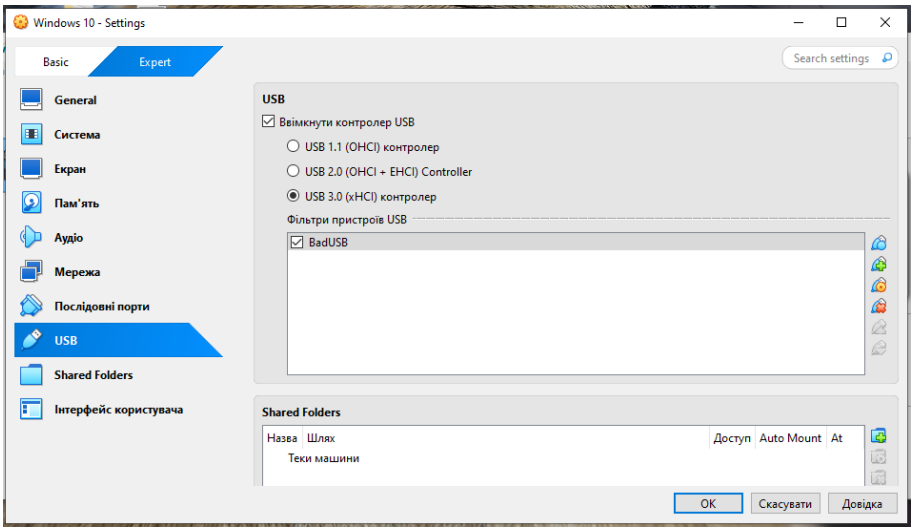

Рисунок 2.3 – Збереження створеного фільтру USB-пристроїв

Тепер при під’єднанні USB-пристроїв вони автоматично спрямовуються до віртуальної машини.

Після запуску віртуальної машини завантажується Arduino IDE.

У браузері Microsoft Edge відкривається сайт www.arduino.cc (див. рис. 2.4).

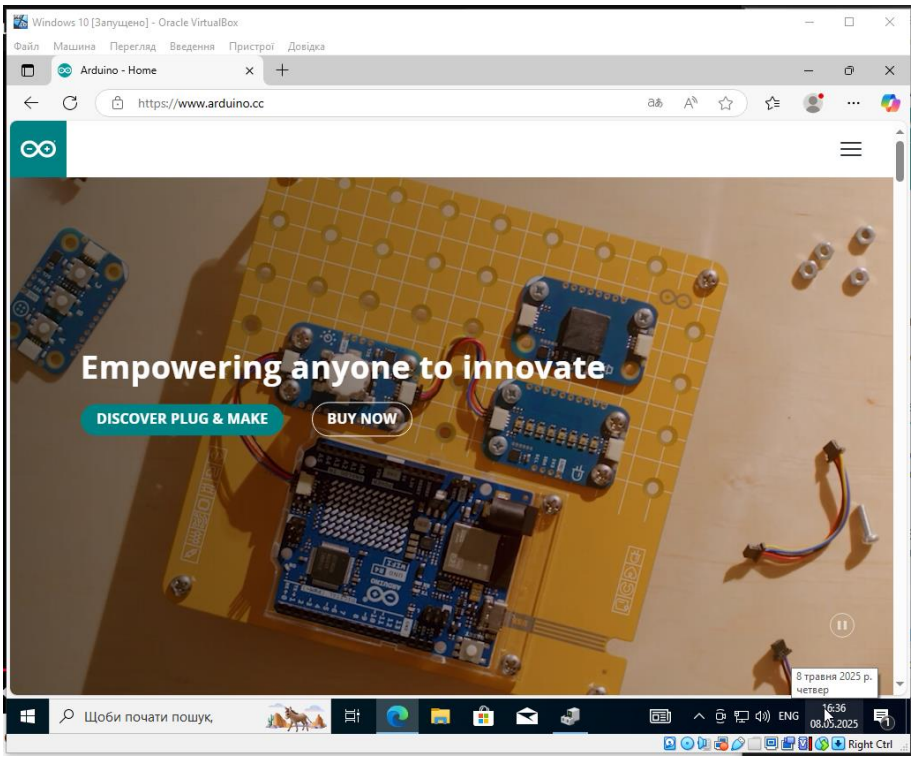

Рисунок 2.4 – Офіційний сайт Arduino

Переходять до вкладки Downloads та обирається пункт «Windows Win 10 and newer, 64 bits» (див. рис. 2.5).

Рисунок 2.5 – Вкладка Downloads на сайті Arduino

Завантаженому інсталятору надається дозвіл на запуск (див. рис. 2.6).

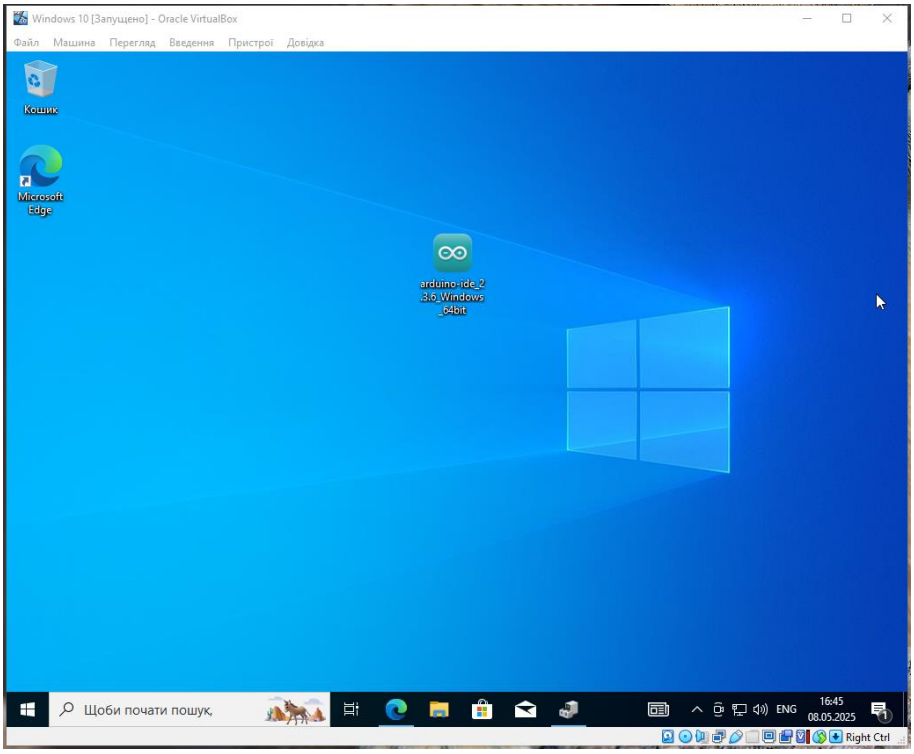

Рисунок 2.6 – Завантажений інсталятор Arduino IDE

Це підтверджується відповідним запитом системи (див. рис. 2.7).

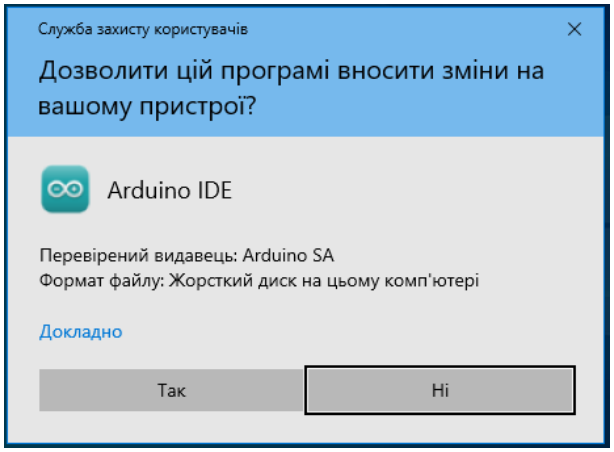

Рисунок 2.7 – Надання дозволу на запуск інсталятора Arduino IDE

Приймається ліцензійна угода (див. рис. 2.8).

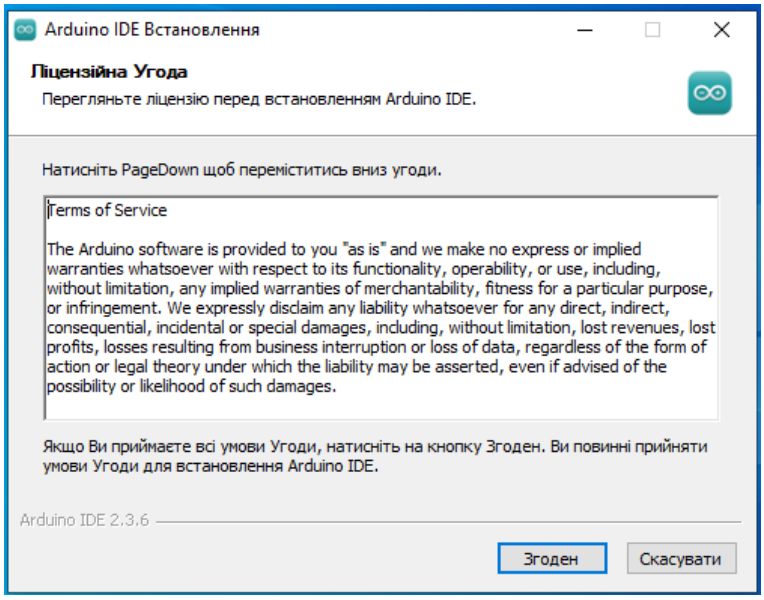

Рисунок 2.8 – Прийняття ліцензійної угоди перед встановленням Arduino IDE

Обирається опція «Anyone who uses this computer (all users)» (див. рис. 2.9).

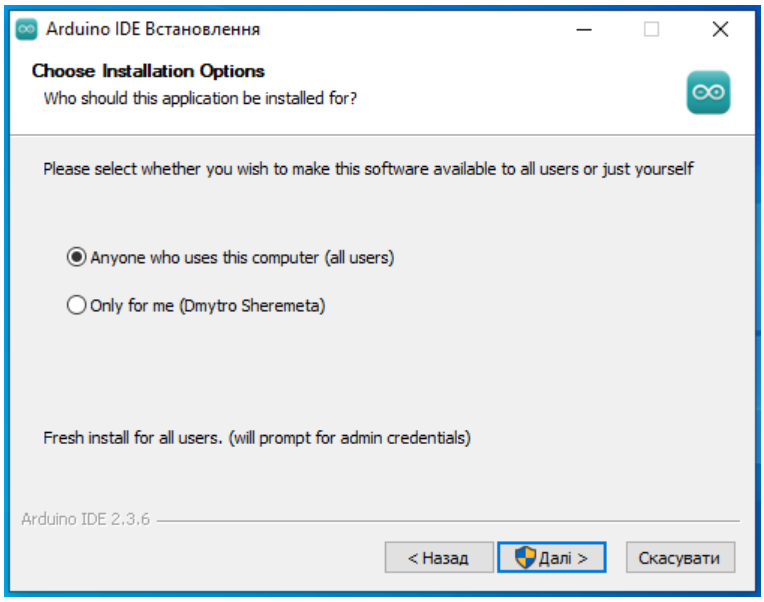

Рисунок 2.9 – Вибір користувачів, які зможуть користуватись Arduino IDE

Зазначається каталог для встановлення та натискається кнопка «Встановити» (див. рис. 2.10).

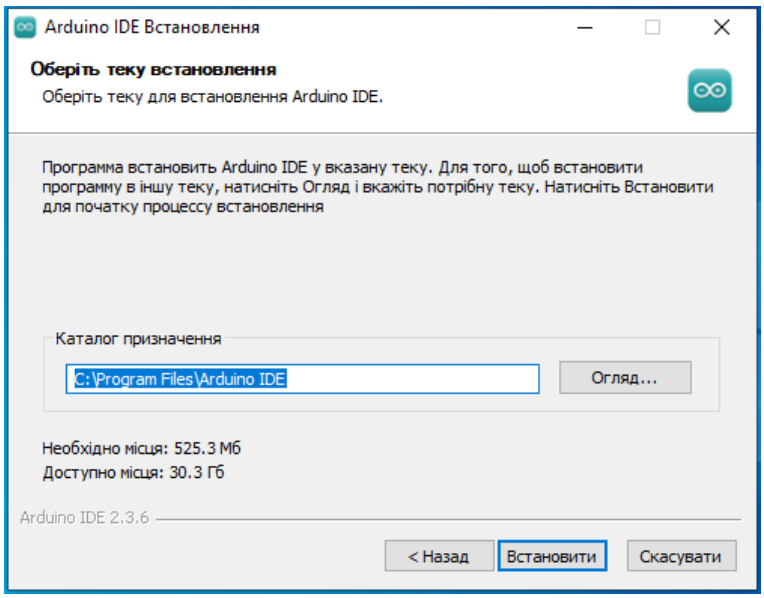

Рисунок 2.10 – Вибір каталогу для встановлення Arduino IDE

Після завершення інсталяції натискається «Кінець» (див. рис. 2.11).

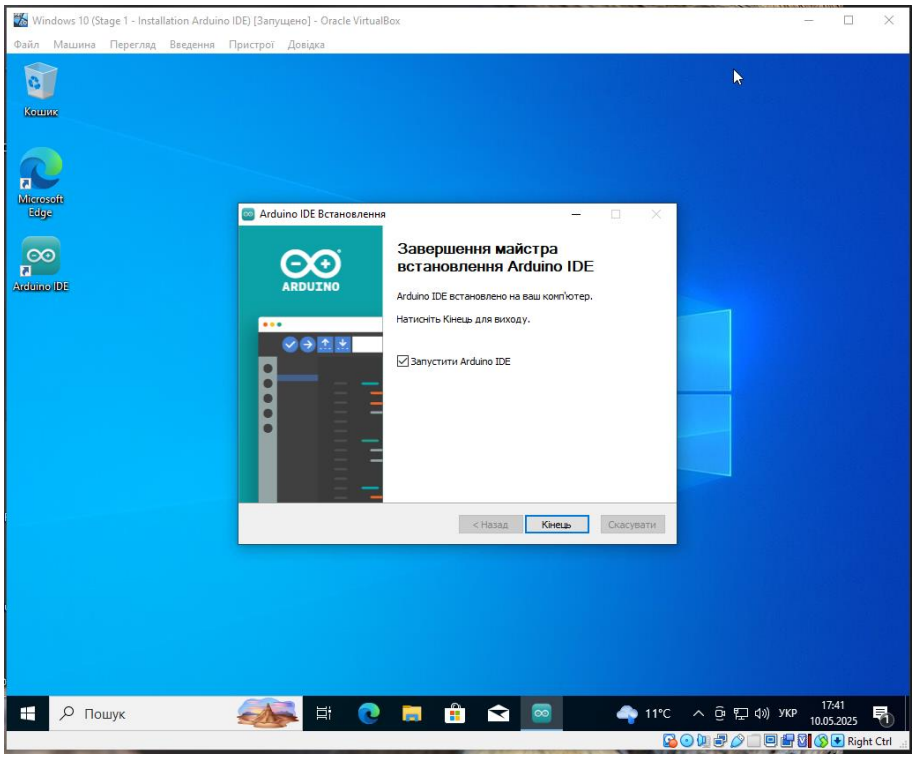

Рисунок 2.11 – Вікно з повідомленням про успішне встановлення Arduino IDE

Після запуску IDE надається дозвіл на доступ до мереж для програм Arduino IDE (див. рис. 2.12).

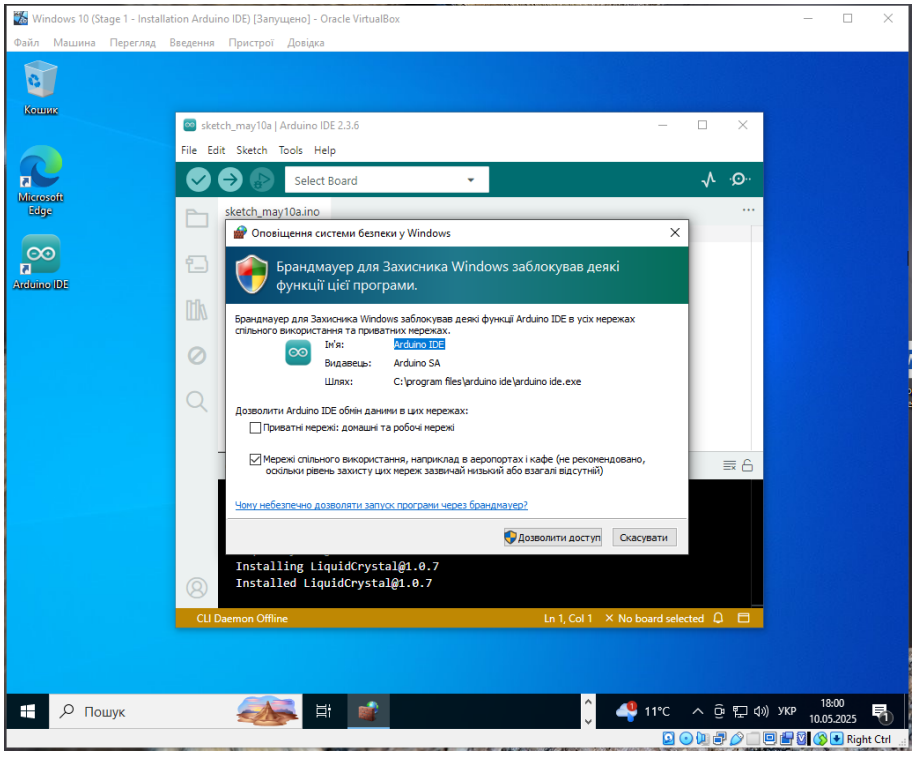

Рисунок 2.12 – Надання дозволу на доступ до мереж спільного використання для програми Arduino IDE

Аналогічний дозвіл надається для «mdns-discovery» (див. рис. 2.13).

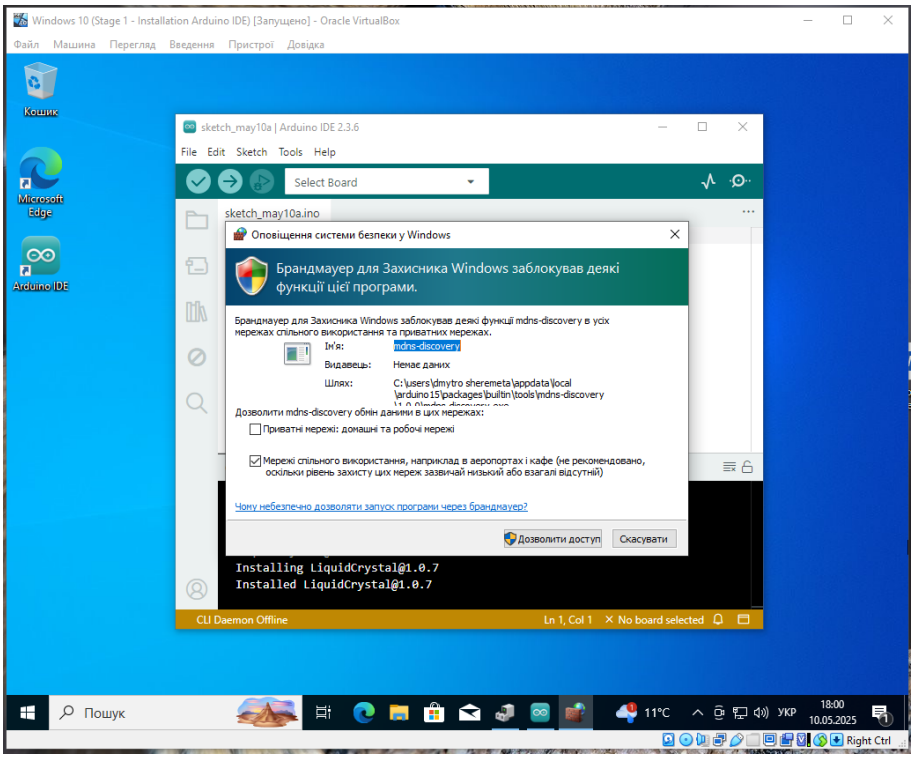

Рисунок 2.13 – Надання дозволу на доступ до мереж спільного використання для програми «mdns-discovery»

Після завершення налаштувань усе готово для написання скетчу для BadUSB-пристрою (див. рис. 2.14).

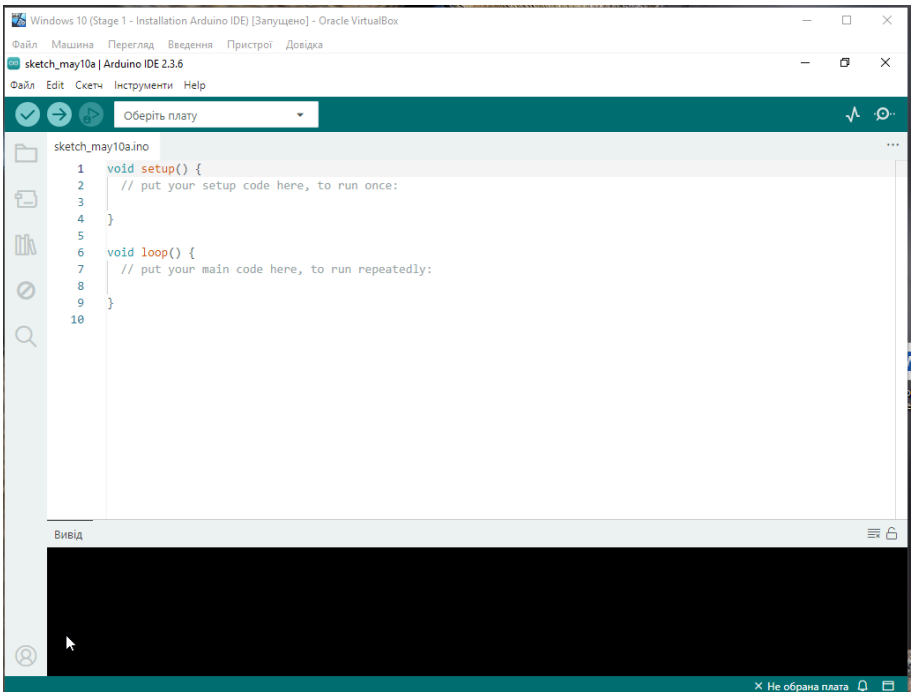

Рисунок 2.14 – Готове до роботи середовище Arduino IDE

Пристрій під’єднується до фізичного USB-порту хоста, зазначеного раніше. У диспетчері пристроїв віртуальної машини він відображається в категорії «Контролери універсальної послідовної шини» як USB Composite Device, а в категорії «Порти (COM та LPT)» – як USB Serial Device (COM3) (див. рис. 2.15).

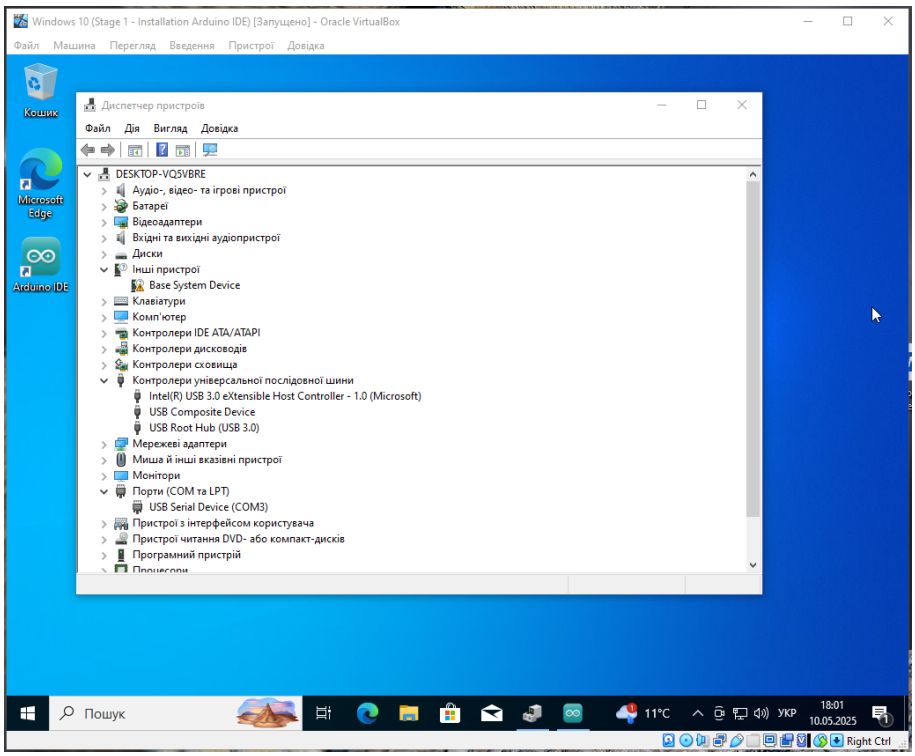

Рисунок 2.15 – Відображення BadUSB-пристрою в диспетчері пристроїв

Позначення *USB Composite Device* свідчить, що пристрій підтримує кілька USB-функцій одночасно. У цьому випадку Arduino Micro є складеним USB-пристроєм, здатним одночасно виконувати функції емуляції клавіатури та серійного порту.

USB Serial Device (COM3) вказує на використання віртуального COM-порту з номером COM3, який призначено для завантаження прошивки (скетчу) і моніторингу даних через інструмент Serial Monitor у середовищі Arduino IDE.

Перед створенням скетчу доцільно розглянути поняття fork-бомби та її роль у BadUSB-атаках. Fork-бомба – це шкідливий код, що створює нескінченну кількість процесів, часто шляхом рекурсивного виклику. У PowerShell, наприклад, така атака може реалізовуватись нескінченним запуском процесу

cmd.exe. У результаті операційна система витрачає всі доступні ресурси (центральний процесор, оперативну пам’ять), що призводить до її зависання або автоматичного перезавантаження. Цей тип атаки часто реалізується через BadUSB-пристрої, які імітують клавіатуру та вводять шкідливі команди автоматично.

Для створення скетчу в Arduino IDE попередньо обирається підключена плата та відповідний COM-порт. У вкладці «Оберіть плату» у випадаючому списку автоматично визначається пристрій «Arduino Micro COM3» (див. рис. 2.16).

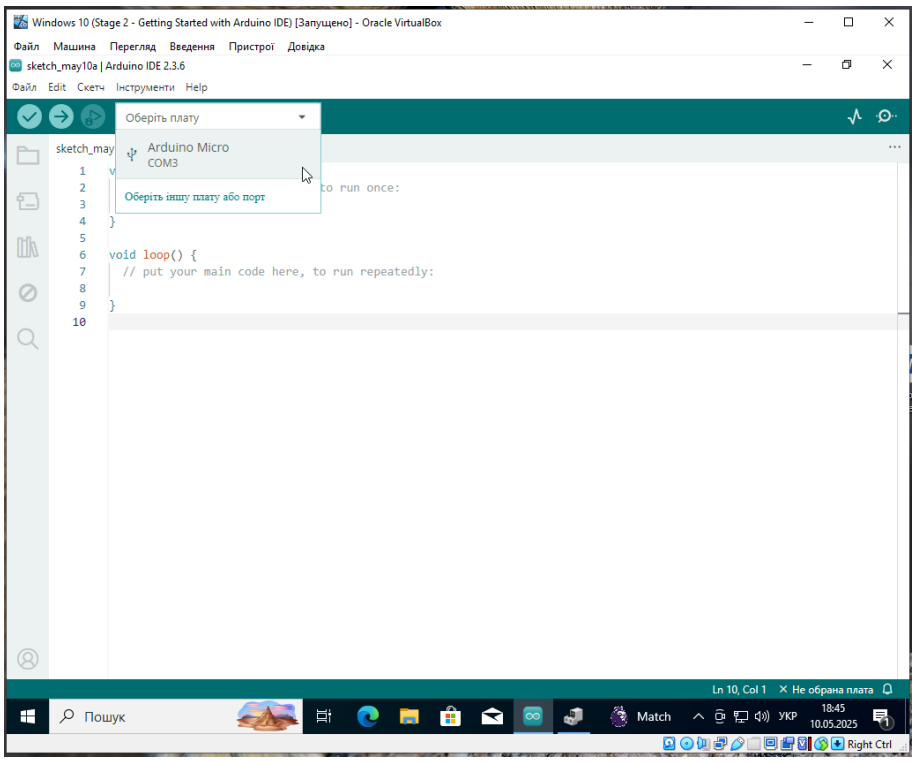

Рисунок 2.16 – Вибір підключеної плати «Arduino Micro COM3»

Скетч, що створюється, виконує такі дії:

– запускає PowerShell через командний рядок (cmd);

– приховує вікно PowerShell;

– перевіряє ім’я комп’ютера: якщо воно дорівнює «Elephant», виконання команд припиняється;

– у протилежному випадку виконується шкідлива команда PowerShell, яка ініціює fork-бомбу.

Після написання скетчу натискається кнопка з піктограмою галочки («Перевірити») для компіляції коду (див. рис. 2.17).

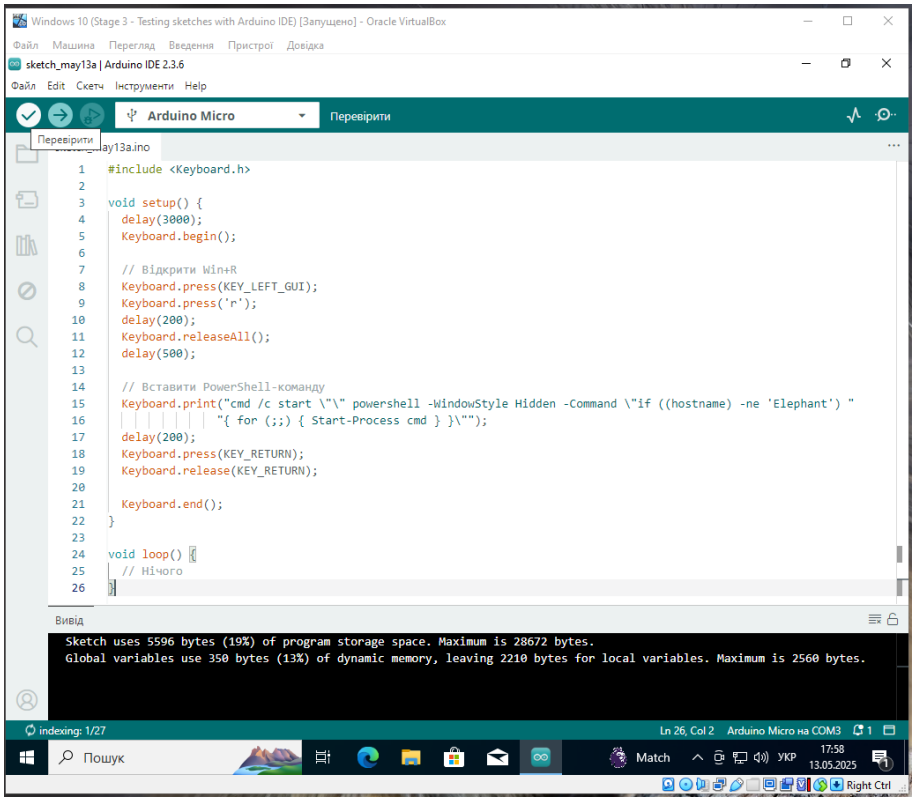

Рисунок 2.17 – Перевірка та компіляція скетчу

У нижній частині вікна Arduino IDE відображається стандартне інформаційне повідомлення з результатами компіляції, що містить дані про використання пам’яті, стан Flash-пам’яті та оперативної пам’яті. У даному випадку ресурси є достатніми для успішного виконання скетчу.

Далі натискається кнопка зі стрілкою вправо («Завантажити») для передавання скетчу до мікроконтролера (див. рис. 2.18).

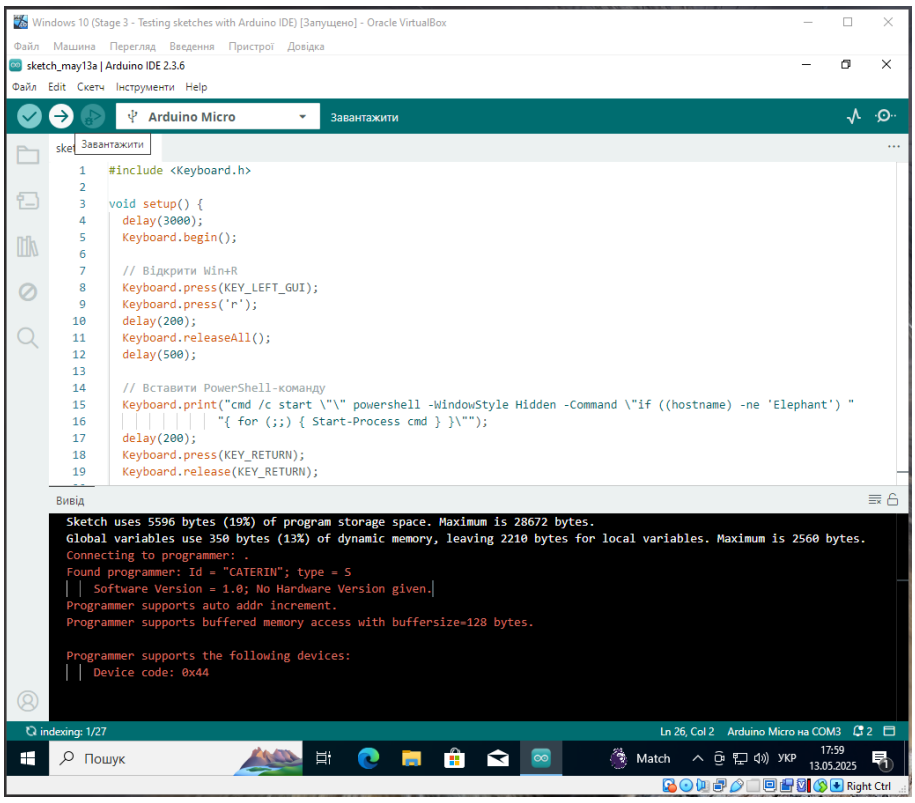

Рисунок 2.18 – Завантаження скетчу на BadUSB-пристрій

Arduino IDE автоматично розпізнає плату Arduino Micro із завантажувачем Caterina та повідомляє про готовність до завантаження. Це свідчить про те, що скетч успішно записано на пристрій.

Після завантаження прошивки BadUSB-пристрій автоматично перепідключається до системи та виконує скетч. У цьому випадку відбувається перевірка імені комп’ютера, і залежно від результату – шкідлива команда або виконується, або ні. Такий підхід дає змогу безпечно використовувати пристрій на власному ПК із заданим іменем комп’ютера, а також змінювати прошивку без ризику автоматичного запуску атаки.

При підключенні пристрою до віртуальної машини скетч ініціює виконання fork-бомби (див. рис. 2.19).

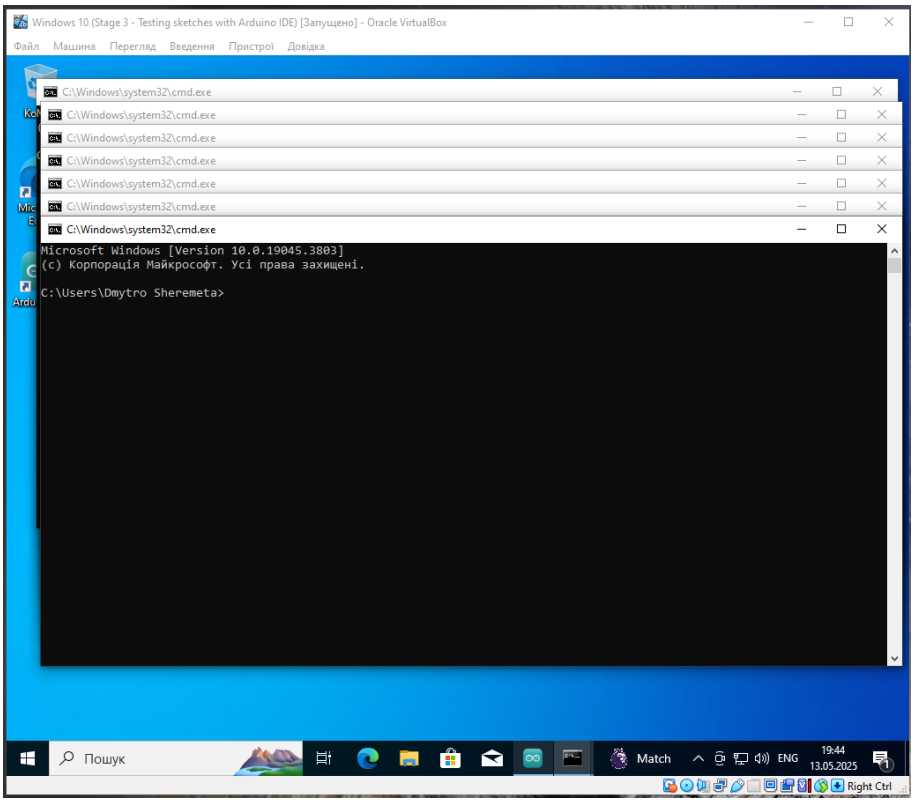

Рисунок 2.19 – Запущена fork-бомба на віртуальній машині

Вона перевантажує ресурси операційної системи (див. рис. 2.20), що в підсумку призводить до її зупинки.

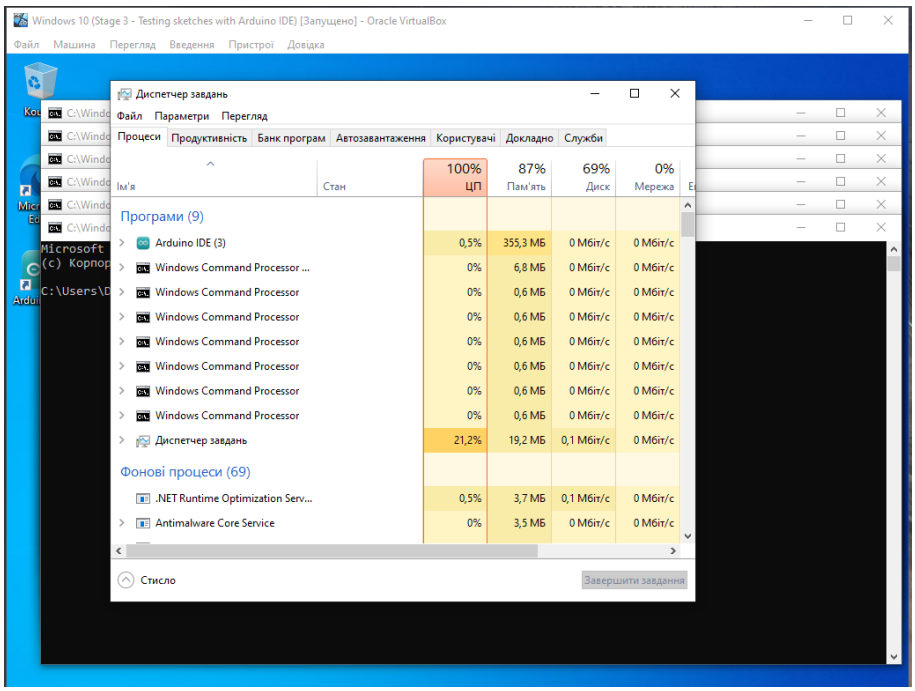

Рисунок 2.20 – Відображення в диспетчері задач перевантажених ЦП та оперативної пам’яті

Скетч спрацює лише за ідеальних умов. Оскільки пристрій працює як клавіатура, він не має доступу до внутрішньої інформації про систему, зокрема мову введення чи стан клавіш. Якщо активна розкладка – не англійська, команда може бути набрана з помилками. Також увімкнений Caps Lock може порушити синтаксис команд, що призведе до їх невиконання.

У сучасних BadUSB-пристроях ці недоліки враховують – наприклад, попередньо перемикають розкладку клавіатури або перевіряють стан Caps Lock. Але реалізація таких пристроїв значно складніша, потребує додаткових ресурсів і глибших технічних знань.

Висновки за розділом 2

У цьому розділі було розглянуто питання створення та налаштування BadUSB-пристрою на основі мікроконтролера Arduino Micro, що дозволяє реалізовувати атаки типу HID-емуляції. На основі вивченого матеріалу встановлено, що подібні пристрої мають значний потенціал як для тестування інформаційної безпеки, так і для здійснення реальних кібератак. Стан питання у сфері BadUSB-атак свідчить про активне використання таких технологій як у дослідницьких цілях, так і у шкідливих сценаріях, що обумовлює необхідність вивчення їхніх механізмів дії.

У ході дослідження було розв’язано завдання зі створення прототипу BadUSB-пристрою, здатного автономно запускати команди в операційній системі Windows через емуляцію введення з клавіатури. Було встановлено, що Arduino Micro з завантажувачем Caterina коректно розпізнається як USB Composite Device і USB Serial Device, що дозволяє легко інтегрувати пристрій у систему. Завантаження скетчу з реалізацією fork-бомби через Arduino IDE здійснюється успішно, а сам пристрій виконує відповідний код при підключенні до системи, що підтверджує працездатність обраного підходу.

Метод вирішення задачі полягав у використанні Arduino IDE для написання скетчу, який автоматично запускає PowerShell, приховує його вікно,

перевіряє ім’я комп’ютера та виконує шкідливу команду лише за відповідної умови. Такий підхід забезпечує контрольований запуск атаки та можливість безпечного тестування пристрою на комп’ютері розробника. Порівняно з іншими варіантами реалізації HID-атак, дане рішення є простим у реалізації, не вимагає спеціалізованого програмного забезпечення чи високих технічних навичок, проте має обмеження, пов’язані з мовою введення, станом клавіш Caps Lock тощо.

Достовірність результатів підтверджено фактичним виконанням команди у віртуальній машині, що призводить до зупинки ОС внаслідок перевантаження ресурсів. Практичне значення отриманих результатів полягає у демонстрації дієвості HID-атак та необхідності впровадження захисту від подібних загроз в корпоративних IT-системах.

Рекомендовано використовувати отримані результати для навчальних та дослідницьких цілей у сфері інформаційної безпеки, а також для моделювання сценаріїв потенційних атак при проведенні тестів на проникнення.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ МЕХАНІЗМУ ЗАХИСТУ ВІД АТАК ТИПУ BADUSB

3.1 Розгляд механізмів захисту

Атаки типу BadUSB становлять серйозну загрозу для інформаційної безпеки підприємств, оскільки базуються на зміні прошивки USB-пристроїв з метою їх несанкціонованого використання – зокрема, для емуляції клавіатури, запуску шкідливого коду або отримання несанкціонованого доступу до системи. Надійна протидія таким загрозам потребує комплексного підходу, що включає як організаційні, так і технічні заходи та засоби захисту.

Першим і одним із найважливіших заходів є навчання працівників. Атаки BadUSB часто базуються на людському факторі – наприклад, підключення працівником «знайденого» флеш-накопичувача або невідомого USB-пристрою до робочого комп’ютера. Навчання повинно охоплювати такі аспекти [19]:

– розпізнавання ознак потенційно шкідливих USB-пристроїв;

– розуміння того, що навіть на вигляд звичайний накопичувач може виконувати шкідливі дії;

– ознайомлення з політикою безпечного використання зовнішніх носіїв;

– вміння оперативно реагувати на підозрілу поведінку системи після підключення USB-пристрою.

Регулярне проведення тренінгів та внутрішніх тестів (наприклад, за допомогою контрольованих атак-перевірок) сприяє формуванню навичок відповідальної поведінки.

Технічною основою захисту є системи програмного контролю USB-портів. Існують різноманітні засоби, які дозволяють:

– блокувати автоматичне розпізнавання нових USB-пристроїв;

– дозволяти лише попередньо авторизовані пристрої (за серійним номером, Vendor ID (VID)/ Product ID (PID), або цифровим підписом);

– встановлювати політики доступу за типом пристрою (дозволяти лише миші, клавіатури, принтери тощо);

– вести журнал усіх спроб підключення нових пристроїв.

До таких засобів належать, наприклад, USB Device Control від Endpoint Protector [20], USBGuard, а також вбудовані засоби ОС (наприклад, групові політики Windows або AppArmor у Linux). Деякі з них також дозволяють реалізовувати "білий список" USB-пристроїв.

Ще одним рівнем захисту є апаратні USB-фільтри, що функціонують як проміжна ланка між комп’ютером та підключеним USB-пристроєм. Вони можуть:

– блокувати небажані класи пристроїв (наприклад, HID);

– забезпечувати електричну розв'язку для захисту від шкідливого обладнання.

Для перевірки підозрілих USB-пристроїв доцільно використовувати спеціальні ізольовані середовища (англ. Sandbox), в яких запуск команд або емуляція клавіатури не матиме наслідків для основної IT-інфраструктури. Такий підхід дозволяє безпечно протестувати поведінку пристрою перед його використанням у робочому середовищі.

Системи захисту кінцевих точок (англ. Endpoint Detection and Response) дозволяють фіксувати аномальні дії, які можуть бути пов’язані з BadUSB-атаками, наприклад:

– раптовий запуск сценаріїв PowerShell чи командного рядка;

– підозріла активність у файловій системі;

– неочікувані мережеві з’єднання після підключення пристрою.

Для захисту на фізичному рівні доцільно:

– обмежити доступ до USB-портів (наприклад, блокування портів заглушками);

– впровадити політику заборони використання власних USB-пристроїв;

– використовувати корпоративні флешки з апаратним шифруванням;

– забезпечити візуальний контроль за підключенням нових пристроїв у робочих зонах.

Таким чином, дієвий захист від атак BadUSB вимагає багаторівневого підходу, що включає як технічні, так і організаційні заходи. Поєднання інструментів контролю, моніторингу, навчання персоналу та політик фізичної безпеки забезпечує найкращу стійкість до таких атак.

3.2 Netwrix Endpoint Protector

Netwrix Endpoint Protector – це комплексне програмне забезпечення для захисту кінцевих точок, яке забезпечує контроль над використанням зовнішніх пристроїв та реалізує функції Data Loss Prevention (DLP). DLP – це система, призначена для запобігання витоку конфіденційної інформації за межі організації, і в складі Netwrix Endpoint Protector вона дозволяє ефективно виявляти, моніторити та блокувати несанкціоновану передачу даних через зовнішні носії або мережеві канали [21].

Раніше відоме як CoSoSys Endpoint Protector, це рішення дозволяє організаціям здійснювати централізовану політику безпеки щодо доступу до USB-портів, знімних накопичувачів та інших пристроїв введення/виведення. Netwrix Endpoint Protector підтримує багатоплатформенне середовище – Windows, macOS та Linux – і забезпечує гнучке управління через веб-інтерфейс.

Принцип роботи Netwrix Endpoint Protector полягає в постійному моніторингу та контролі підключення периферійних пристроїв до комп’ютерів у мережі організації. Програма дозволяє створювати політики доступу для конкретних типів пристроїв, брендів, моделей або навіть серійних номерів, що особливо важливо у контексті захисту від атак типу BadUSB. Наприклад, USB-пристрій, який видає себе за клавіатуру, але не є авторизованим, буде автоматично заблокований або помічений як підозрілий.

Функціональні можливості системи включають:

– контроль доступу до портів USB та інших інтерфейсів введення/виведення;

– авторизація дозволених пристроїв за серійними номерами;

– журналювання всіх спроб підключення пристроїв;

– можливість створення політик на рівні груп користувачів або окремих комп’ютерів;

– виявлення та блокування невідомих HID-пристроїв;

– централізоване управління з веб-консолі;

– інструменти для аудиту та відповідності вимогам стандартів безпеки.

Netwrix Endpoint Protector є дієвим рішенням саме проти атак типу BadUSB, де шкідливий USB-пристрій імітує поведінку клавіатури або миші з метою автоматичного виконання команд без відома користувача. Завдяки можливості ідентифікувати тип пристрою, перевіряти його ідентифікатори та обмежувати доступ, система дозволяє заблокувати такі несанкціоновані пристрої ще до того, як вони почнуть взаємодіяти з системою. Також важливо, що Netwrix Endpoint Protector не просто блокує USB-порти, а дозволяє гнучко управляти ними, що робить його зручним навіть у великих корпоративних мережах.

Це рішення буде особливо корисним для середніх і великих організацій, де використання знімних пристроїв є поширеним, а ризики, пов’язані з BadUSB, можуть мати суттєві наслідки для інформаційної безпеки. Netwrix Endpoint Protector дозволяє ефективно запобігати інцидентам, пов’язаним з фізичним доступом до комп’ютерів, що особливо актуально в умовах соціальної інженерії або внутрішніх загроз.

3.3 Реалізація захисту від атаки BadUSB

Розгорнуто віртуальну машину з програмним забезпеченням Netwrix Endpoint Protector. Endpoint Protector надає програмне забезпечення як для серверної частини – центрального вузла, що керує політиками безпеки, так і для

клієнтських комп'ютерів – кінцевих точок. Локальне програмне забезпечення, встановлене на кожній кінцевій точці, називається агентом. Саме агент відповідає за дотримання встановлених політик безпеки на комп'ютерах організації.

Було налаштовано віртуальну машину – сервер Endpoint Protector Appliance. Після запуску віртуальної машини першим кроком стало прийняття ліцензійної угоди для Endpoint Protector Appliance (див. рис. 3.1).

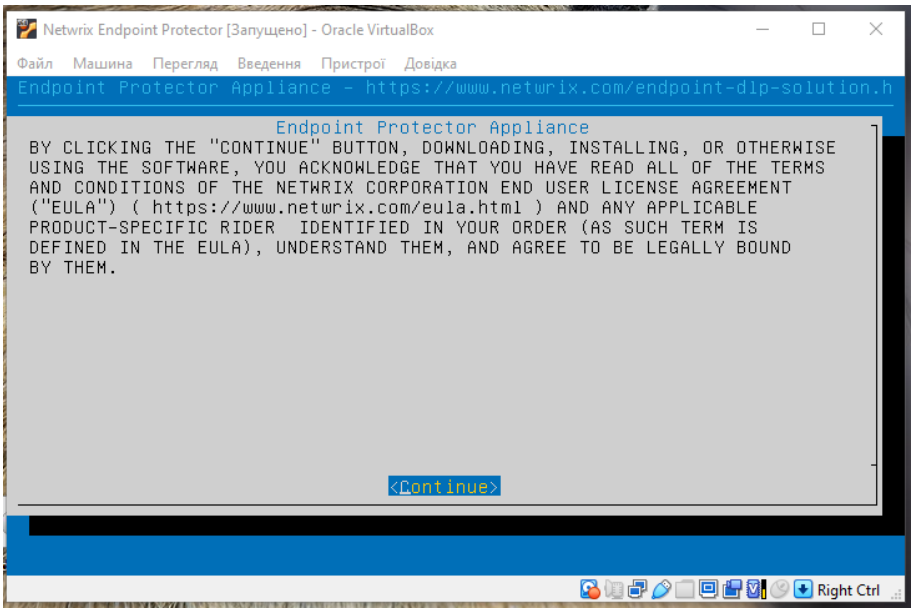

Рисунок 3.1 – Прийняття ліцензійної угоди для Endpoint Protector Appliance

Перед початком роботи з сервером Endpoint Protector Appliance необхідно виконати базові дії, що забезпечують легітимність і відповідність умовам використання програмного забезпечення.

Прийняття ліцензійної угоди є обов'язковою процедурою, яка передує налаштуванню функціональних компонентів системи та доступу до її адміністративного інтерфейсу.

Це також підтверджує ознайомлення адміністратора з умовами експлуатації, правами та обмеженнями, які встановлені виробником.

Далі здійснено початкове налаштування Endpoint Protector Appliance через головне меню (див. рис. 3.2).

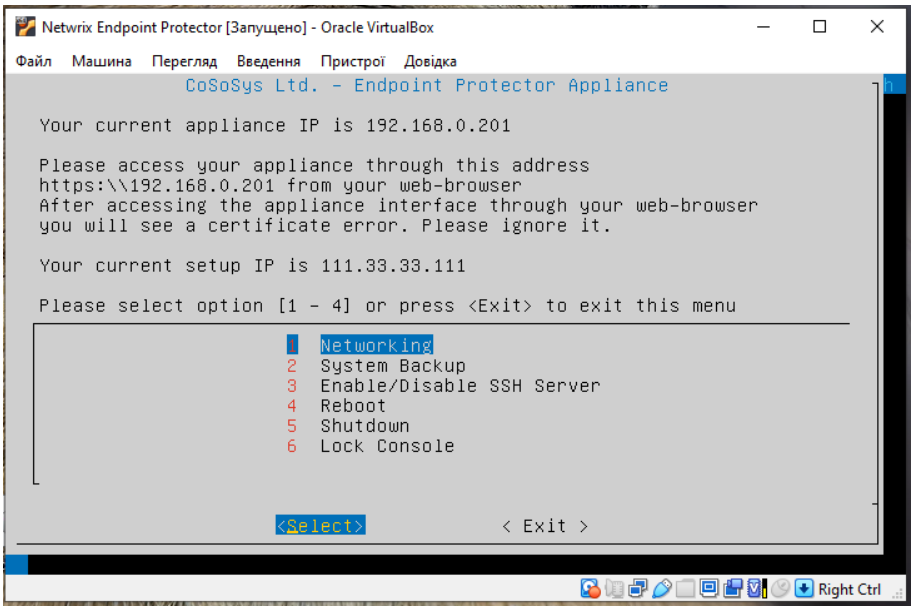

Рисунок 3.2 – Головне меню налаштування Endpoint Protector Appliance

Відображено локальну та зовнішню Internet Protocol адресу (IP-адресу) – відповідно 192.168.0.201 та 111.33.33.111. За замовчуванням залишено всі параметри, система готова до подальшої роботи.

З віртуальної машини з встановленим Arduino IDE у веб-браузері відкрито інтерфейс керування Endpoint Protector за IP-адресою https://192.168.0.201. Для входу до системи використано стандартні облікові дані: логін – root, пароль – epp2011 (див. рис. 3.3).

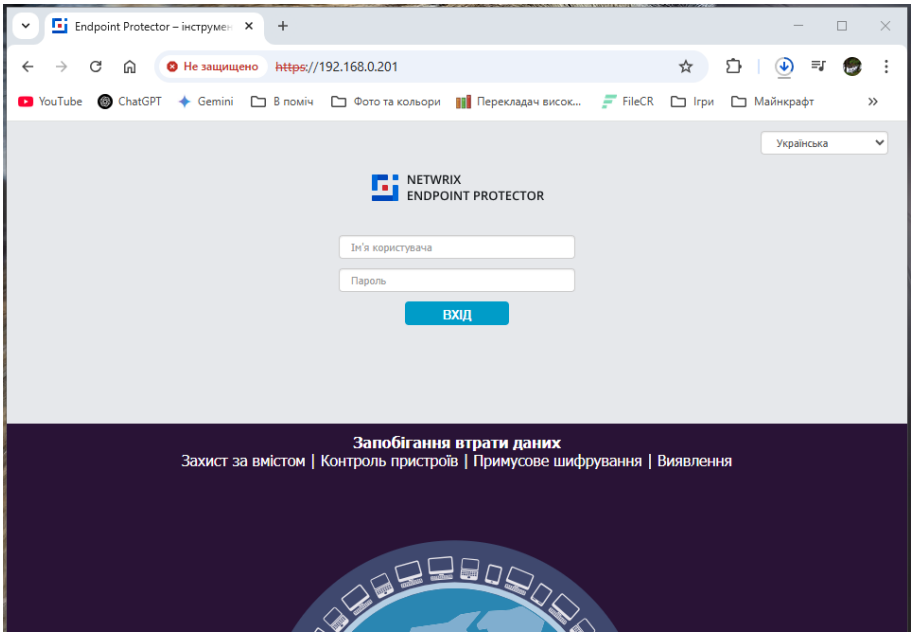

Рисунок 3.3 – Вхід до інтерфейсу керування Endpoint Protector

Виконано налаштування центру керування Endpoint Protector.

Спочатку обрано відповідний часовий пояс (див. рис. 3.4).

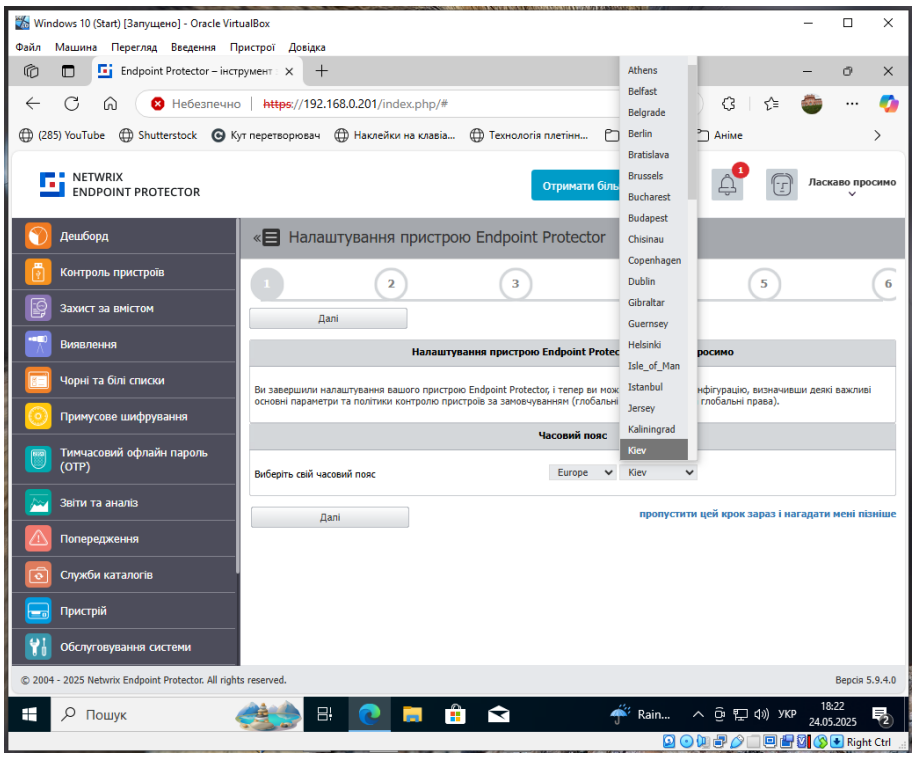

Рисунок 3.4 – Налаштування центру керування Endpoint Protector – обрано відповідний часовий пояс

Правильне встановлення часового поясу є важливим для синхронізації журналів подій, планування політик безпеки та коректного функціонування компонентів системи.

Особливо це актуально в корпоративному середовищі, де сервери та клієнтські пристрої можуть знаходитися в різних часових зонах.

Невірно вказаний часовий пояс може призвести до зміщень у часових мітках подій, ускладнити аудит дій користувачів або моніторинг спроб підключення до системи.

Тому цей крок доцільно виконувати на самому початку налаштування центру керування.

Далі – тип ліцензії, наприклад Free Trial (див. рис. 3.5).

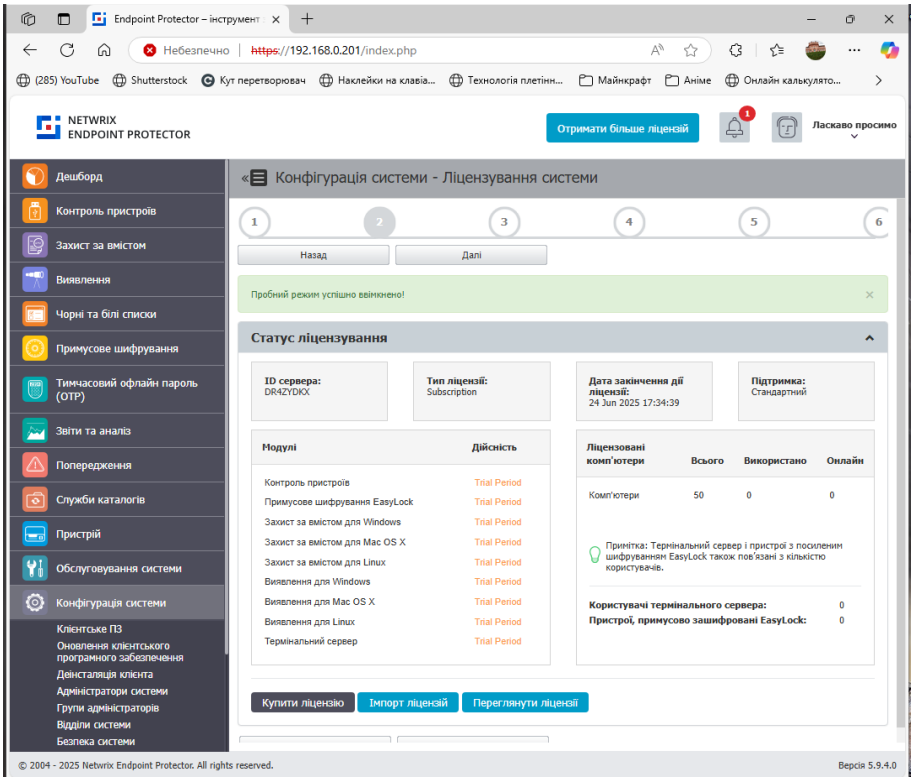

Рисунок 3.5 – Налаштування центру керування Endpoint Protector – обрано тип ліцензії Free Trial

На сторінках майстра налаштування 3, 4 та 5 залишено параметри за замовчуванням, після чого встановлено політику «Заборонити доступ до USB-накопичувачів» і натиснуто «Завершити» (див. рис. 3.6).

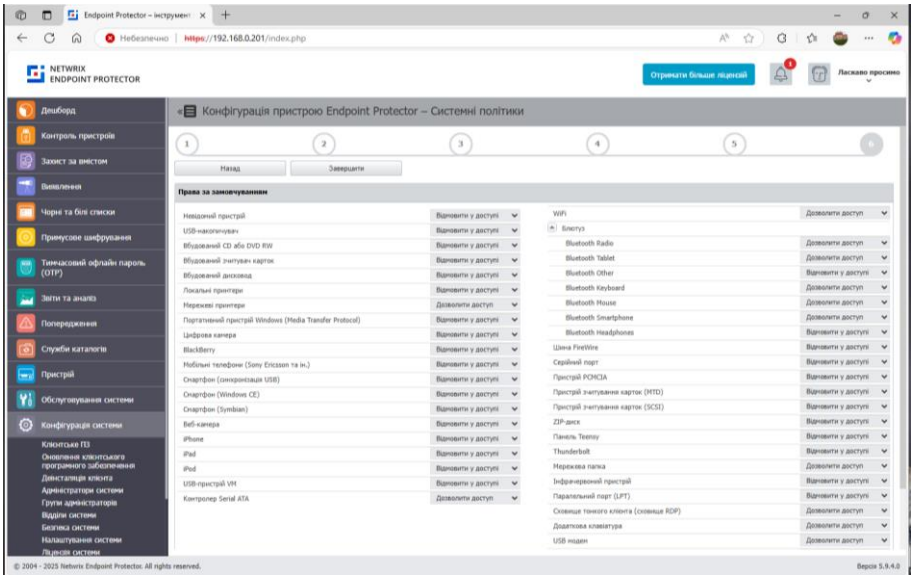

Рисунок 3.6 – Налаштування центру керування Endpoint Protector – встановлено політику «Заборонити доступ до USB-накопичувачів»

На віртуальний комп’ютер встановлено клієнт Endpoint Protector. У веб-інтерфейсі обрано вкладку «Конфігурація системи», розділ «Клієнтське ПЗ» і натиснуто «Завантажити» (див. рис. 3.7).

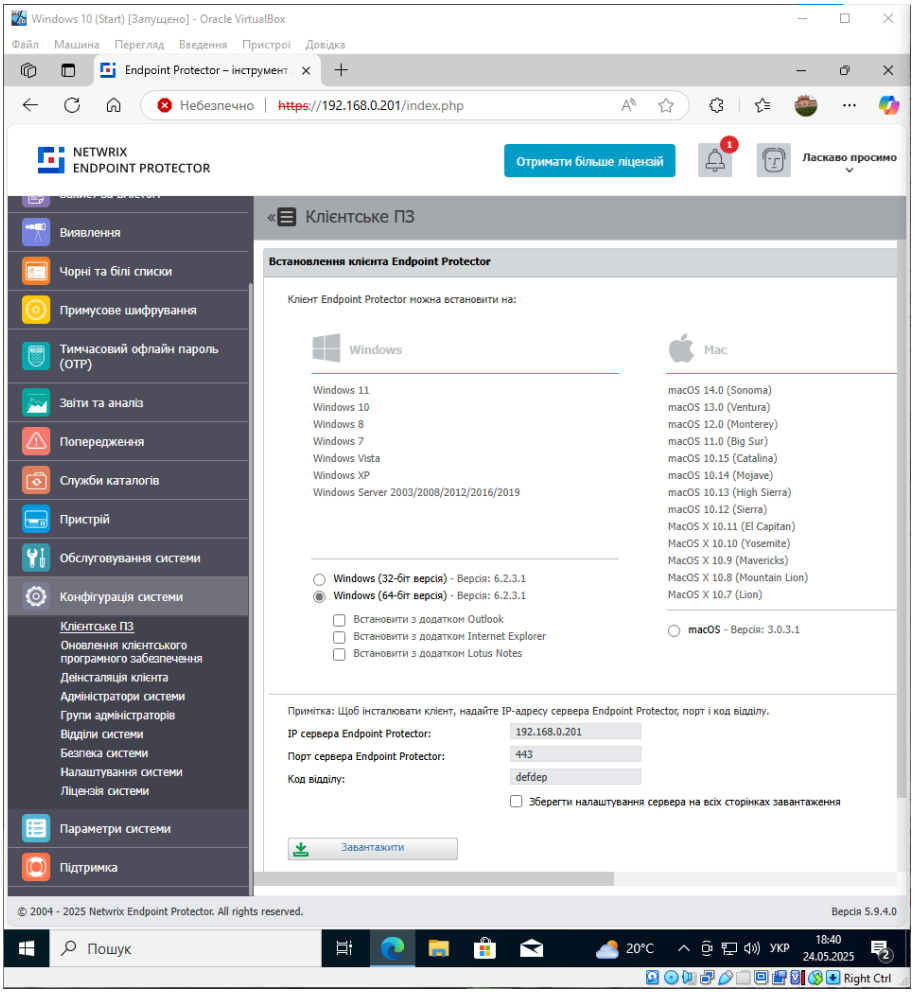

Рисунок 3.7 – Вкладка «Конфігурація системи», розділ «Клієнтське ПЗ», натиснуто «Завантажити»

Процес завантаження клієнтського програмного забезпечення є невід’ємною частиною впровадження системи контролю портів і пристроїв.

Цей крок забезпечує можливість встановлення агента на робочі станції користувачів, що дозволяє здійснювати моніторинг, контроль доступу до USB-портів та виконання політик безпеки, визначених адміністратором.

Інтерфейс управління дозволяє швидко знайти потрібний дистрибутив клієнта, адаптований до версії операційної системи цільового пристрою.

Наявність централізованого механізму завантаження клієнта також полегшує розгортання системи в масштабі підприємства.

Запущено інсталятор завантаженого програмного забезпечення (див. рис. 3.8).

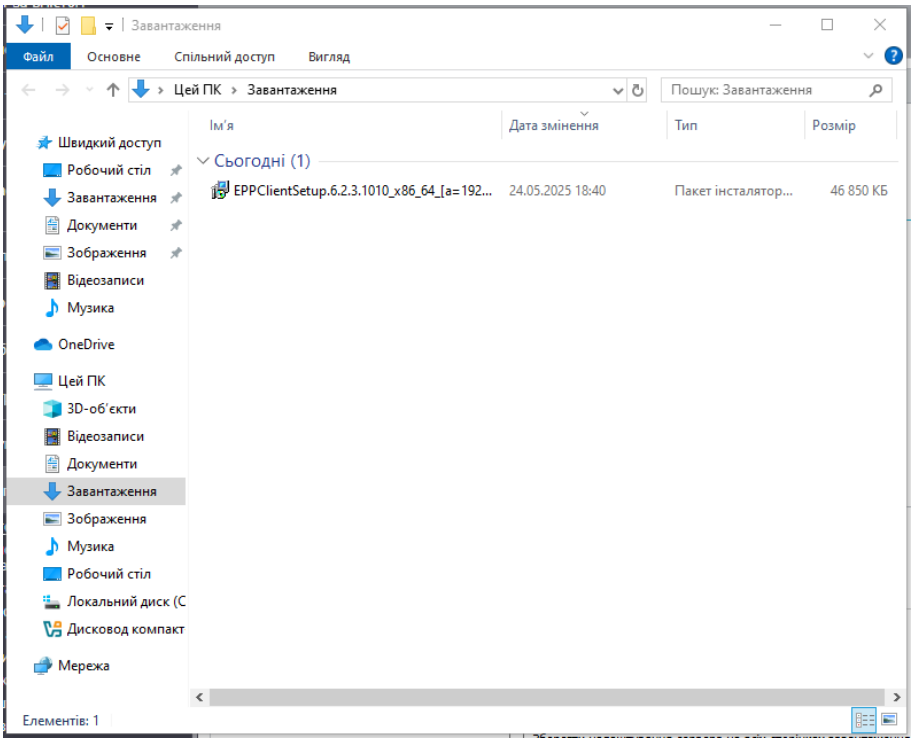

Рисунок 3.8 – Інсталятор завантаженого клієнтського програмного забезпечення

У вікні встановлення натиснуто «Next» (див. рис. 3.9).

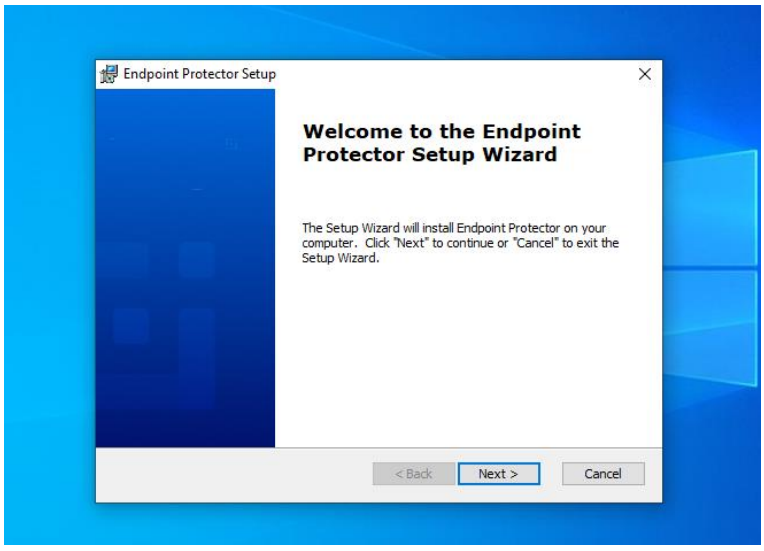

Рисунок 3.9 – Вікно встановлення клієнтського ПЗ, натиснуто «Next»

Обрано місце встановлення програми (див. рис. 3.10).

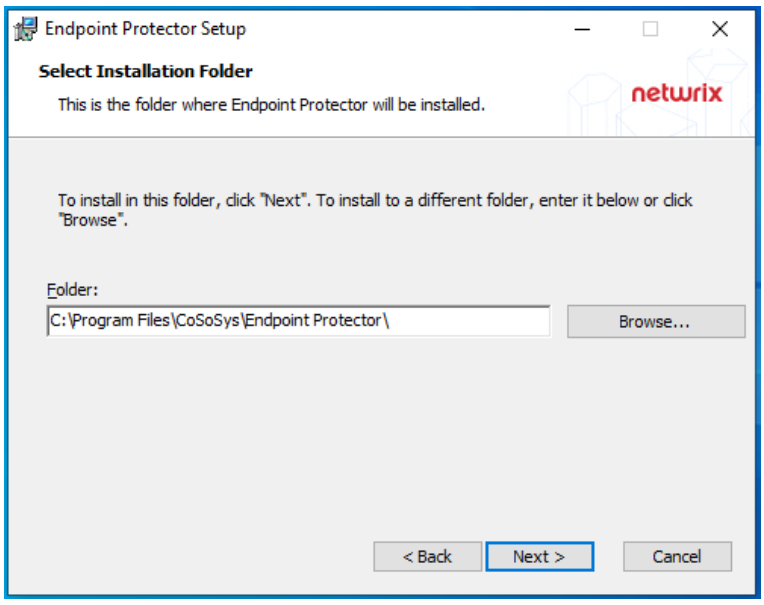

Рисунок 3.10 – Вікно встановлення клієнтського ПЗ, обрано місце встановлення програми

Вказано IP-адресу сервера, порт та код департаменту (див. рис. 3.11).

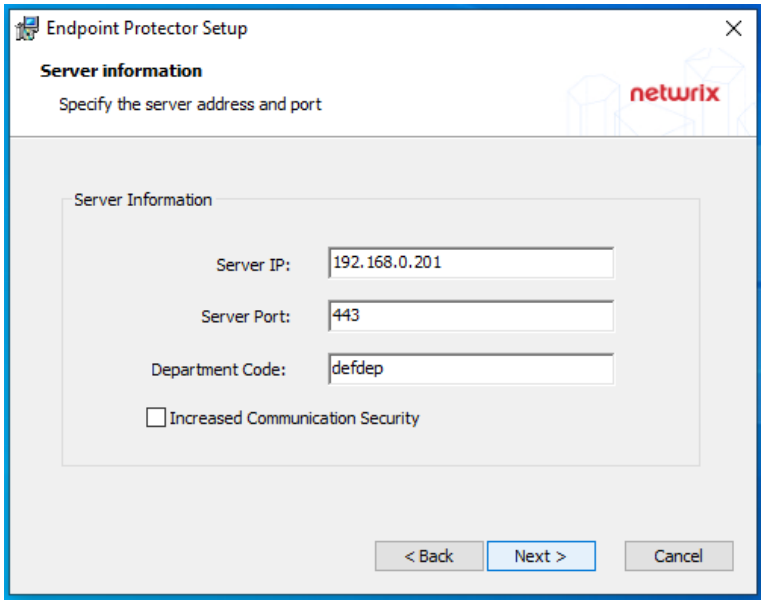

Рисунок 3.11 – Вікно встановлення клієнтського ПЗ, вказано IP-адресу сервера, порт та код департаменту

Обрано системні параметри (див. рис. 3.12).

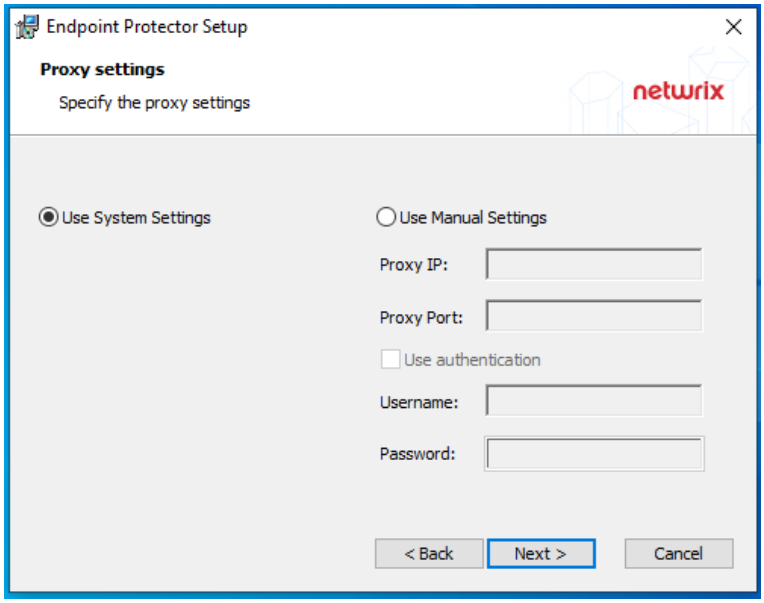

Рисунок 3.12 – Вікно встановлення клієнтського ПЗ, обрано системні параметри

Натиснуто «Install» (див. рис. 3.13).

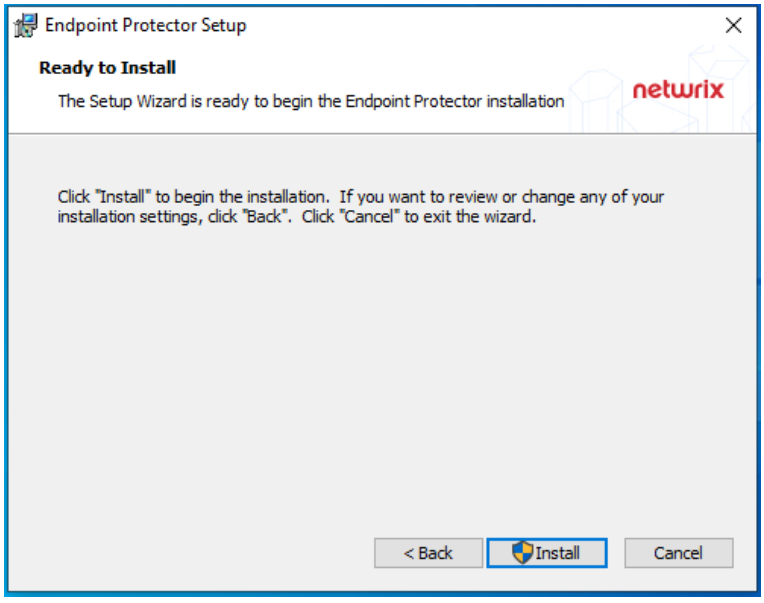

Рисунок 3.13 – Вікно встановлення клієнтського ПЗ, натиснуто «Install»

Після завершення встановлення підтверджено дозвіл на внесення змін у систему (див. рис. 3.14).

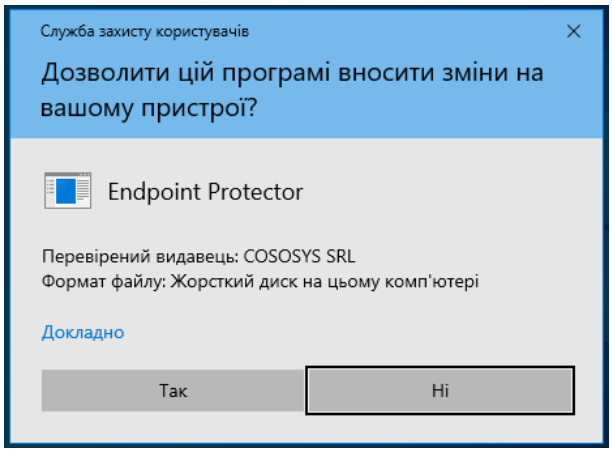

Рисунок 3.14 – Підтверджено дозвіл на внесення змін у систему

Цей етап є стандартною процедурою в операційних системах сімейства Windows і спрямований на захист від несанкціонованого встановлення або змін у системі.

Підтвердження внесення змін свідчить про завершення інсталяційного процесу клієнтського програмного забезпечення та його інтеграцію в операційну систему.

Також це вказує на успішне завершення попередніх кроків інсталяції й готовність системи до наступних налаштувань або перевірки з'єднання з сервером керування.

Натиснуто «Finish» (див. рис. 3.15).

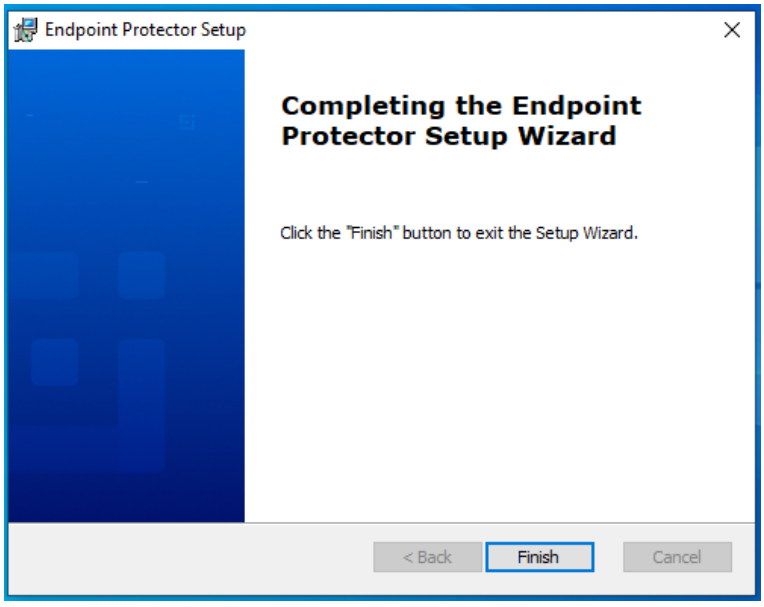

Рисунок 3.15 – Вікно встановлення клієнтського ПЗ, натиснуто «Finish»

Після встановлення клієнтського програмного забезпечення у веб-інтерфейсі, у вкладці «Контроль пристроїв», у розділі «Комп’ютери» відображено комп’ютер у списку пристроїв, підключених до сервера (див. рис. 3.16).

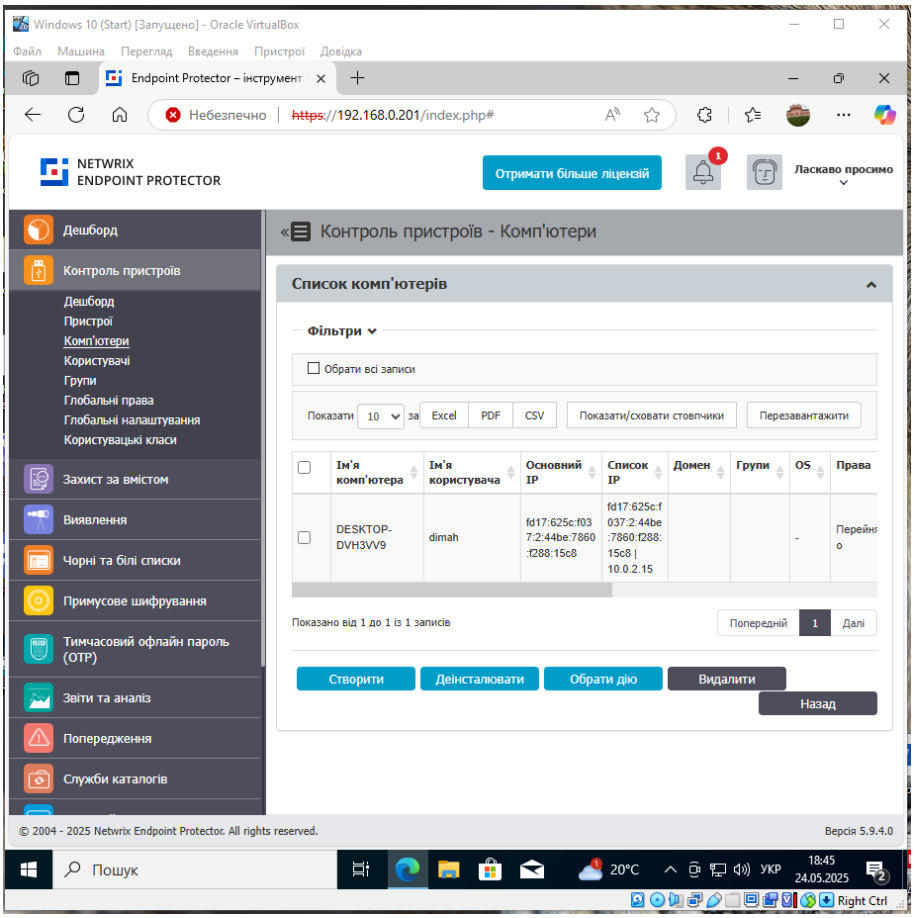

Рисунок 3.16 – Вкладка «Контроль пристроїв», розділ «Комп’ютери», відображено комп’ютер у списку пристроїв, підключених до сервера

Відображення клієнтського пристрою у списку підключених комп’ютерів свідчить про успішне встановлення агента та його коректну взаємодію з сервером Endpoint Protector.

Це підтверджує, що клієнтська частина системи змогла ініціалізувати зв’язок із сервером, передати ідентифікаційну інформацію та готова до застосування налаштованих політик безпеки.

Такий механізм є основою централізованого контролю в корпоративному середовищі.

Під час спроби підключити USB-флеш-накопичувач з’явилося сповіщення про блокування пристрою (див. рис. 3.17).

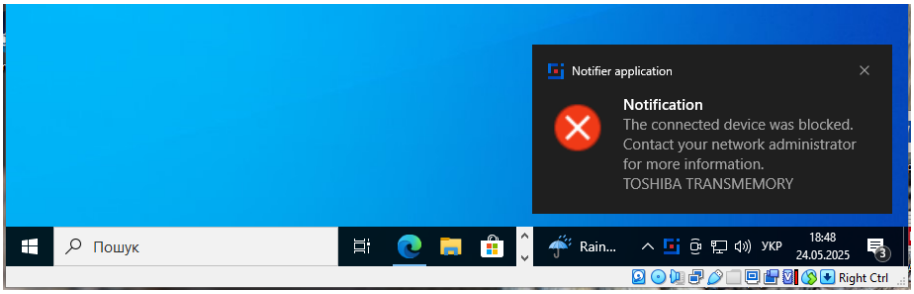

Рисунок 3.17 – Сповіщення про блокування USB-флеш-накопичувача

Таким чином, підключення сторонніх USB-накопичувачів заблоковано відповідно до налаштованої політики.

При під’єднанні BadUSB-пристрою було виведено сповіщення про блокування пристрою *Microsoft USB Serial Device (COM3)*, а також відкрито вікно командного рядка, де спроба виконання команди українською мовою завершилася невдало (див. рис. 3.18).

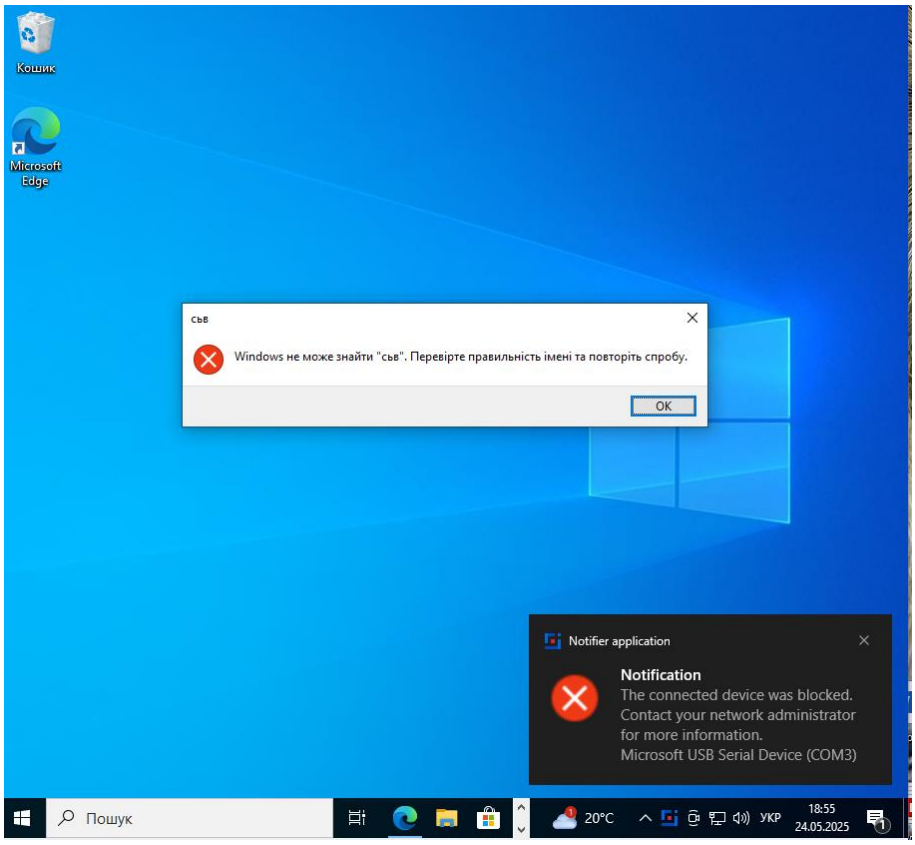

Рисунок 3.18 – Сповіщення про блокування пристрою Microsoft USB Serial Device (COM3) та відкрите вікно командного рядка

Оскільки BadUSB-пристрій на базі Arduino Micro визначається як USB Composite Device, він ідентифікується як клавіатура та як USB-серійний пристрій одночасно. Серійний пристрій було заблоковано, однак клавіатура залишилася активною. У зв’язку з цим було виявлено необхідність доопрацювання політики блокування.

Для внесення змін у веб-інтерфейсі Endpoint Protector відкрито вкладку «Контроль пристроїв», обрано розділ «Глобальні права» (див. рис. 3.19).

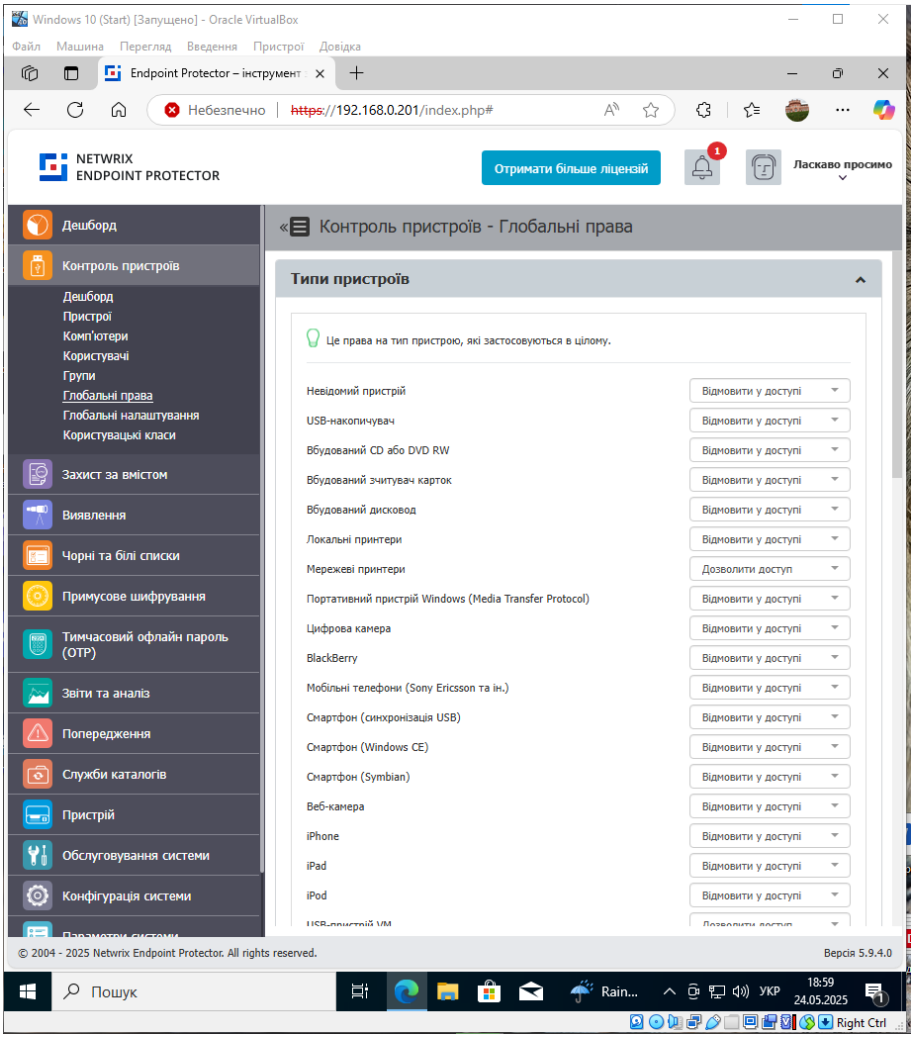

Рисунок 3.19 – Вкладка «Контроль пристроїв», розділ «Глобальні права»

У нижній частині сторінки знайдено тип пристрою «Додаткова клавіатура» та встановлено політику «Відмовити у доступі» (див. рис. 3.20). Зміни збережено.

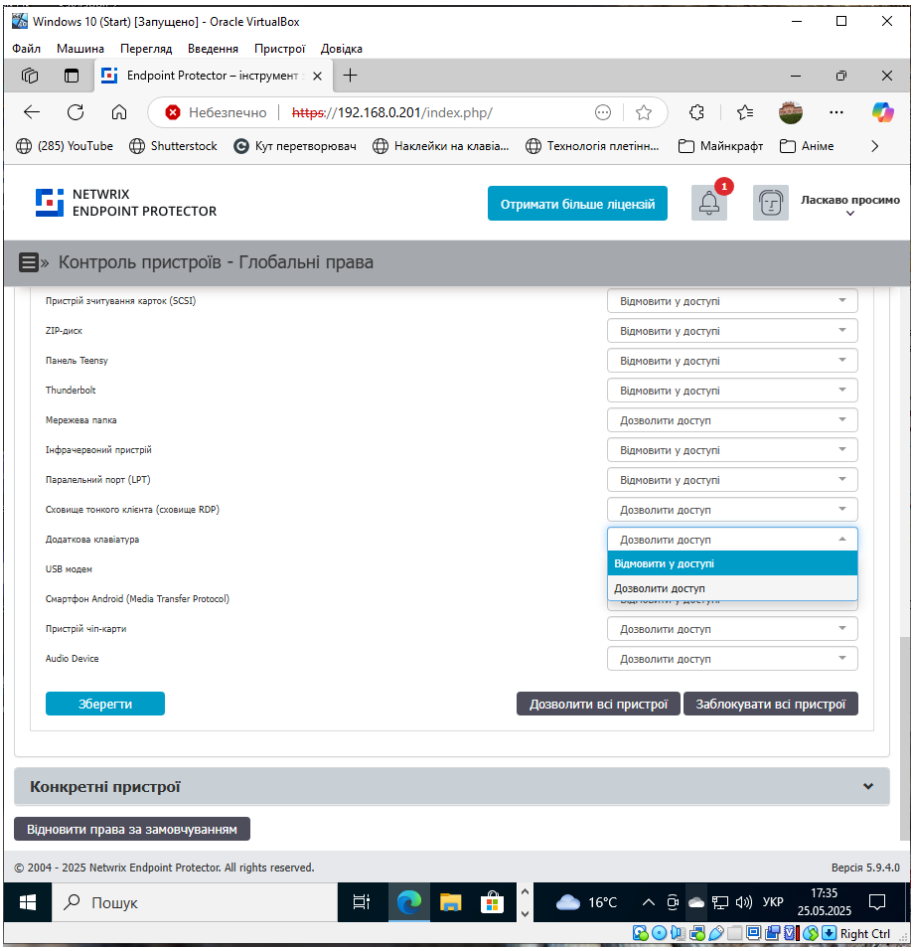

Рисунок 3.20 – Вкладка «Контроль пристроїв», розділ «Глобальні права», встановлено політику «Відмовити у доступі» типу пристрою «Додаткова клавіатура»

Після повторного під’єднання BadUSB-пристрою відображено два окремих сповіщення:

про блокування клавіатури (див. рис. 3.21);

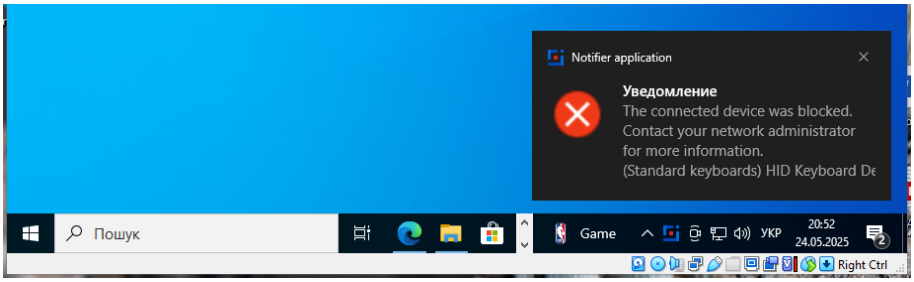

Рисунок 3.21 – Сповіщення про блокування клавіатури

про блокування серійного пристрою (див. рис. 3.22).

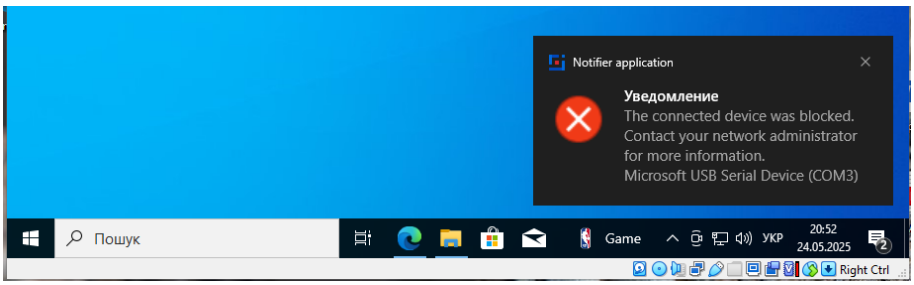

Рисунок 3.22 – Сповіщення про блокування пристрою Microsoft USB Serial Device (COM3)

Це підтверджує коректну роботу Endpoint Protector та захищеність комп’ютера від сторонніх USB-пристроїв.

На завершення перевірено список пристроїв, що під’єднувалися до комп’ютера протягом роботи Endpoint Protector. Для цього відкрито вкладку «Контроль пристроїв» і обрано розділ «Пристрої» (див. рис. 3.23).

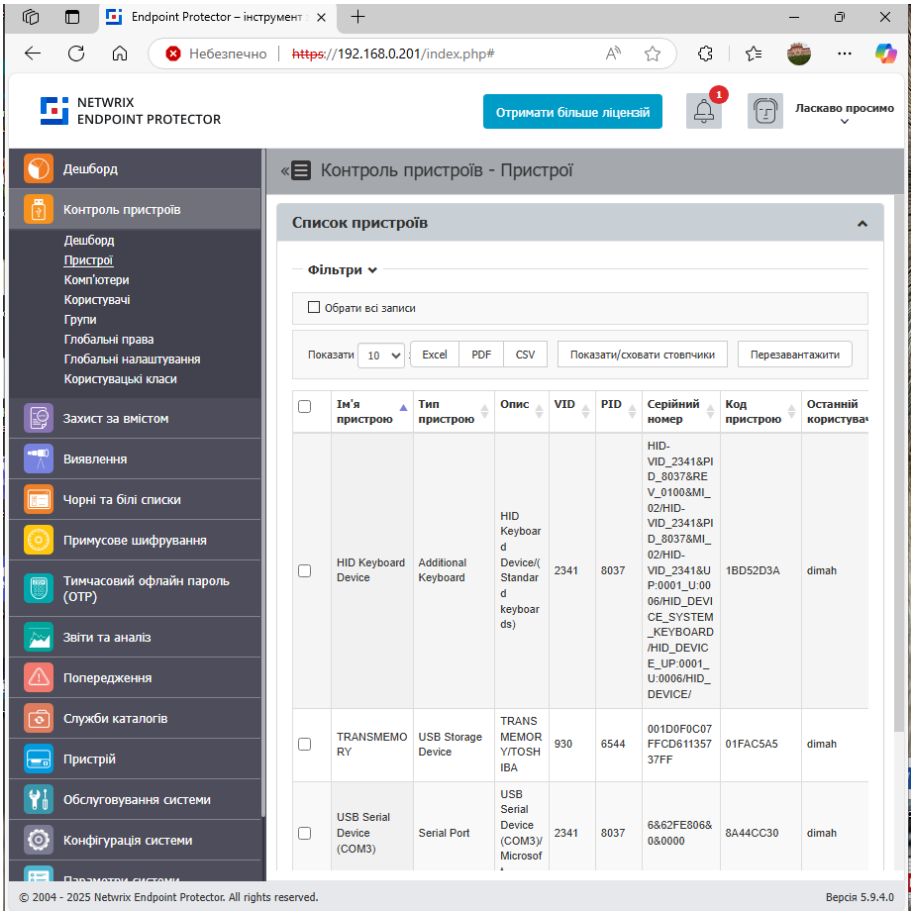

Рисунок 3.23 – Вкладка «Контроль пристроїв», розділ «Пристрої», список пристроїв, що під’єднувалися до комп’ютера протягом роботи Endpoint Protector

3.4 Рекомендації щодо захисту від атак типу BadUSB

Захист інформаційних систем від атак типу BadUSB потребує комплексного підходу, що поєднує організаційні заходи, апаратні та програмні засоби. Дієва протидія таким загрозам передбачає не лише застосування технологічних рішень, а й формування належної політики безпеки та підвищення обізнаності працівників. У зв’язку з цим доцільним є формування практичних рекомендацій щодо побудови надійної системи захисту від атак BadUSB з урахуванням сучасних засобів, зокрема Netwrix Endpoint Protector.

Рекомендації щодо захисту від атак типу BadUSB:

1. Організаційні заходи

– Розробка та впровадження політики безпечного використання USB-пристроїв. У ній слід передбачити заборону підключення неавторизованих пристроїв, порядок використання корпоративних накопичувачів, а також відповідальність за порушення правил.

– Підвищення обізнаності працівників. Регулярно проводити навчання та інструктажі з виявлення потенційно небезпечних USB-пристроїв. Особливу увагу приділяти ознакам емуляції клавіатури та іншим проявам шкідливої активності.

– Проведення внутрішніх перевірок. Застосовувати контрольовані тестові атаки для перевірки готовності працівників до реагування на загрози BadUSB.

– Обмеження використання особистих носіїв. Запровадити заборону на використання приватних USB-пристроїв у корпоративному середовищі.

2. Апаратні засоби

– Фізичне блокування USB-портів. Використовувати спеціальні заглушки або відключати порти через BIOS/UEFI на комп’ютерах, де USB-пристрої не потрібні.

– Використання USB-фільтрів. Встановлювати апаратні засоби, які блокують певні класи пристроїв (наприклад, HID) або забезпечують електричну розв’язку з підозрілими пристроями.

– Контроль за підключенням. Організувати відеонагляд або призначити відповідальних осіб для моніторингу підключення зовнішніх пристроїв у критичних зонах.

– Використання захищених корпоративних носіїв. Надати працівникам сертифіковані USB-накопичувачі з апаратним шифруванням.

3. Програмні засоби

– Впровадження системи контролю USB-пристроїв. Таким сучасним інструментом є Netwrix Endpoint Protector, який дозволяє:

- створювати "білий список" дозволених пристроїв за vid/pid або серійним номером;
- блокувати неавторизовані підключення та небезпечні класи пристроїв (наприклад, hid);
- вести журнал усіх спроб підключення;
- реалізовувати гнучкі політики доступу до USB-портів за типом пристрою, користувачем або групою.

– Використання засобів виявлення аномальної активності. Інтегрувати системи EDR, які відслідковують раптовий запуск скриптів, зміну файлів чи неочікувані мережеві з'єднання після підключення USB.

– Ізольоване тестування пристроїв. Підозрілі USB варто перевіряти у віртуалізованому або sandbox-середовищі, що не має доступу до критичних ресурсів підприємства.

Висновки за розділом 3

Проблема захисту від атак типу BadUSB набула особливої актуальності в умовах зростання загроз, пов’язаних із фізичним доступом до пристроїв,

використанням скомпрометованих USB-носіїв і неможливістю виявлення таких атак традиційними антивірусними або мережевими засобами. Це обумовлює потребу у впровадженні комплексних заходів захисту, що охоплюють організаційні, апаратні та програмні компоненти.

Для вирішення поставленої задачі було сформовано практичні рекомендації, які передбачають підвищення обізнаності працівників, запровадження політик контролю доступу до USB-портів, використання апаратних фільтрів, а також створення ізольованих середовищ для тестування підозрілих пристроїв. Ключовим програмним компонентом реалізації технічних заходів стало програмне забезпечення Netwrix Endpoint Protector, яке забезпечує контроль пристроїв та DLP-функціональність.

У процесі практичної реалізації було розгорнуто серверну й клієнтську частини Netwrix Endpoint Protector у віртуальному середовищі, налаштовано політики блокування USB-накопичувачів та HID-пристроїв. Проведене тестування з використанням BadUSB-пристрою на базі Arduino Micro засвідчило дієвість системи: після налаштування політик було забезпечено повне блокування як серійного інтерфейсу, так і клавіатурного емулятора. Це доводить здатність системи адаптуватися до нестандартних USB-загроз.

Таким чином, впровадження Netwrix Endpoint Protector дозволяє не лише запобігти BadUSB-атакам, але й комплексно підвищити рівень безпеки кінцевих точок шляхом централізованого управління, аудитів і гнучкого налаштування політик доступу до апаратних ресурсів.

ВИСНОВКИ

У кваліфікаційній роботі було проведено комплексне дослідження проблеми атак типу BadUSB, їх технічної реалізації та можливих механізмів захисту. Аналіз наукових джерел, практичних прикладів та експериментальна реалізація дозволили сформулювати низку важливих висновків:

1. Було проаналізовано актуальність загроз, пов’язаних із використанням BadUSB-пристроїв, які становлять серйозну небезпеку через здатність маскуватися під легітимні USB-пристрої, водночас виконуючи приховані шкідливі дії. Виявлено ключові вразливості USB-інтерфейсу, зокрема відсутність автентифікації пристроїв, перевірки їхньої цілісності та можливість перепрошивки мікроконтролерів, що дозволяє змінювати функціональність пристрою без фізичного втручання. Досліджено архітектуру BadUSB на базі мікроконтролерів із підтримкою HID (зокрема ATmega32U4), які дають змогу емуляції клавіатури або інших пристроїв введення. Також було вивчено методи створення таких пристроїв із використанням як готових рішень (USB Rubber Ducky), так і користувацьких платформ (MalDuino на Arduino), із порівнянням їх функціональних можливостей, мов програмування та підходів до реалізації шкідливих сценаріїв.

2. У ході дослідження було класифіковано основні види атак типу BadUSB за функціональним призначенням: емуляційні, завантажувальні та комбіновані. Емуляційні пристрої імітують інші та виконують заздалегідь запрограмовані команди, наприклад, через PowerShell. Завантажувальні пристрої містять у собі шкідливе програмне забезпечення, яке автоматично встановлюється після підключення. Комбіновані пристрої поєднують обидві функції – введення команд і зберігання шкідливих файлів. Також було проаналізовано реальні сценарії використання BadUSB: атаки на операційні системи, запуск шкідливих скриптів, саботаж (наприклад, через fork-бомбу), крадіжку даних та створення облікових записів з підвищеними правами. Отримані результати дозволили

оцінити реальні наслідки таких атак і підтвердити необхідність впровадження цільових заходів захисту.

3. У результаті дослідження було створено робочий прототип BadUSB-пристрою на базі плати Arduino Micro з мікроконтролером ATmega32U4, що підтримує HID-функції. Було розроблено скетч, який емулює клавіатуру та автоматично виконує заздалегідь запрограмовані команди через PowerShell. Для підвищення контрольованості було реалізовано перевірку імені комп’ютера перед запуском атаки, що дозволяє уникнути випадкової активації на сторонніх пристроях. У віртуалізованому середовищі було протестовано поведінку пристрою, зокрема виконання fork-бомби, яка спричиняє перевантаження ресурсів операційної системи. Отримані результати підтвердили працездатність пристрою, реальність загрози та дієвість автоматизованого сценарію BadUSB-атаки.

4. Практична реалізація захисту включала використання програмного забезпечення Netwrix Endpoint Protector, яке забезпечує контроль над використанням зовнішніх пристроїв та реалізує функції обмеження доступу до USB-інтерфейсів. У межах роботи було налаштовано політику блокування HID-пристроїв, які не входять до білого списку, а також було реалізовано журналювання спроб підключення зовнішніх носіїв. Було проведено тестування захисту на прикладі BadUSB-пристрою, створеного на базі Arduino Micro, що імітував клавіатуру та виконував шкідливі дії. У віртуальному середовищі було перевірено реакцію системи безпеки, яка успішно заблокувала підозріле підключення, що підтвердило дієвість налаштованого технічного захисту.

5. На основі проведеного дослідження було сформовано низку практичних рекомендацій для забезпечення комплексного захисту від BadUSB-атак. До них належать:

– заборона на використання приватних USB-пристроїв у корпоративному середовищі;

– підвищення обізнаності персоналу щодо ризиків BadUSB, а також навчання розпізнаванню потенційних загроз;

– використання апаратних засобів контролю (USB-фільтри, фізичні блокатори портів);

– використання систем контролю пристроїв (наприклад, Netwrix Endpoint Protector);

– створення ізольованих середовищ для перевірки зовнішніх пристроїв.

Отримані результати можуть бути використані:

– у навчальному процесі під час вивчення тем із прикладної кібербезпеки;

– для підготовки лабораторних занять з тематики шкідливих USB-пристроїв;

– під час тестування інформаційної безпеки в IT-інфраструктурах;

– як основа для впровадження політик контролю USB-доступу в корпоративних мережах;

– для підвищення обізнаності користувачів щодо ризиків використання сторонніх USB-пристроїв.

Отже, проведене дослідження дозволило всебічно розкрити особливості реалізації атак з використанням BadUSB, а також запропонувати дієві, комплексні підходи до їх виявлення і блокування. Практична реалізація як атаки, так і механізму протидії підтвердила дієвість обраної методики, а отримані результати можуть слугувати основою для подальших досліджень і розробки політик захисту апаратного рівня IT-систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Вікіпедія*. *USB*. URL: https://uk.wikipedia.org/wiki/USB (дата звернення: 11.12.2024).

2. *Security Research Labs*. *Turning USB peripherals into BadUSB*. URL: https://web.archive.org/web/20160418134155/https:/srlabs.de/badusb/ (дата звернення: 12.12.2024).

3. АТАКИ НА ОСНОВІ BADUSB / В. Корчинський та ін. *Measuring and computing devices in technological processes*. 2023. Т. 4. С. 134–139.

4. Що таке BadUSB - Терміни та визначення у сфері кібербезпеки. *VPN Unlimited - Fast & Secure VPN service*. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/badusb?srsltid=AfmBOor_RHukRPJ5ISS4ZK2BHzUxGUgWTghDeeZinmu92DlRjH_HeT91 (дата звернення: 13.12.2024).

5. BadUSB - What is it and How to Avoid it: Step-by-Step Guide. *Comparitech*. URL: https://www.comparitech.com/net-admin/what-is-badusb/ (дата звернення: 30.12.2024).

6. Перше знайомство з BadUSB. *Arduino в Україні*. URL: https://arduino.ua/art176-pershe-znaiomstvo-z-badusb?srsltid=AfmBOoofRrNaqjDNSKs3vEzIZXdW6ap0KMU0dT6rXDYzuuZZio1vcwdP (дата звернення: 30.12.2024).

7. USB Rubber Ducky. *Hak5*. URL: https://shop.hak5.org/products/usb-rubber-ducky?srsltid=AfmBOorUR86P_ef7OhU_mFehkcE423CQn7Km5EeydnFrSyrSiIfbBYSR (дата звернення: 02.01.2025).

8. Leonardo. *Arduino Documentation*. URL: https://docs.arduino.cc/hardware/leonardo/ (дата звернення: 02.01.2025).

9. Micro. *Arduino Documentation*. URL: https://docs.arduino.cc/hardware/micro/ (дата звернення: 02.01.2025).

10. Плата розробника від Digispark ATtiny85 USB. *Diy Shop*. URL: https://diyshop.com.ua/en/plata-razrabotchika-ot-digispark-attiny85-usb?srsltid=AfmBOooqSG0bJC7xRcuWadoSM6lFu2iSIWkTrT_E5u0L_pXylTSH97d5 (дата звернення: 02.01.2025).

11. The basics. *Maltronics Documentation*. URL: https://docs.maltronics.com/badusb-scripting/the-basics (дата звернення: 05.01.2025).

12. Hello, World!. *Product Documentation*. URL: https://docs.hak5.org/hak5-usb-rubber-ducky/ducky-script-basics/hello-world#macos-example (дата звернення: 05.01.2025).

13. *Arduino - Home*. URL: https://www.arduino.cc/ (дата звернення: 10.01.2025).

14. A BadUSB Device With Arduino. *Instructables*. URL: https://www.instructables.com/A-BadUSB-Device-With-Arduino/ (дата звернення: 13.01.2025).

15. BadUSB. Що це і як захиститись.. *CoreWin*. URL: https://corewin.ua/ceo-blog/what-is-badusb/ (дата звернення: 27.01.2025).

16. MalDuino W. *Maltronics*. URL: https://maltronics.com/collections/malduinos/products/malduino-w?ref=seytonic.com (дата звернення: 27.01.2025).

17. USB Killer. *WhatIs*. URL: https://www.techtarget.com/whatis/definition/USB-Killer (дата звернення: 11.02.2025).

18. ATmega32U4. *Access Denied*. URL: https://www.microchip.com/en-us/product/atmega32u4 (дата звернення: 18.02.2025).

19. Правила Безпечного Користування USB-накопичувачами. *Memory.Net.Ua*. URL: https://memory.net.ua/info/pravila-bezpechnogo-koristuvannja-usb-nakopichuvachami?srsltid=AfmBOorwTW6Ln-iZNhvqtARsU9RGU11dU6UeK3R4nPjc5xzETJSIuZBQ (дата звернення: 16.03.2025).

20. Endpoint Protector Device Control. *Endpoint Protector - Industry-Leading Data Loss Prevention (DLP)*. URL: https://www.endpointprotector.com/solutions/device-control (дата звернення: 01.04.2025).

21. Netwrix Endpoint Protector. *CoreWin*. URL: https://corewin.ua/security/endpoint-protector/ (дата звернення: 03.04.2025).

ДОДАТОК

Реалізація fork-бомби

```
#include <Keyboard.h>

void setup() {
  delay(3000);
  Keyboard.begin();

  // Відкрити Win+R
  Keyboard.press(KEY_LEFT_GUI);
  Keyboard.press('r');
  delay(200);
  Keyboard.releaseAll();
  delay(500);

  // Вставити PowerShell-команду
  Keyboard.print("cmd /c start \"\" powershell -WindowStyle Hidden -Command \"if ((hostname) -ne 'Elephant') "
                 "{ for (;;) { Start-Process cmd } }\"");
  delay(200);
  Keyboard.press(KEY_RETURN);
  Keyboard.release(KEY_RETURN);

  Keyboard.end();
}

void loop() {
  // Нічого
}
```