КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій
(факультет)

інформаційних технологій проектування та прикладної математики
(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

на тему: «Моделювання кіберзагроз при впровадженні Інтернету речей»

СОЛОДЕЙ НІКІТА ІВАНОВИЧ
(прізвище, ім’я та по батькові студента повністю)

Київ 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проектування та прикладної математики

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Терентьєв О.О.

_____
„___” _____2024 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

на тему: «Моделювання кіберзагроз при впровадженні Інтернету речей»

Виконав: студент 2-го курсу, групи ІСТм-23

Спеціальності: 126 «Інформаційні системи та технології» .
(шифр і назва напряму підготовки, спеціальності)

Солодей Н.І. .
(прізвище та ініціали)

Керівник _к.т.н., доц. Горда О.В. .
(прізвище та ініціали)

Рецензент _к.т.н., доц. Шабала Є.Є. .
(прізвище та ініціали)

Київ, 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: автоматизації і інформаційних технологій
.
Кафедра: інформаційних технологій проектування та ПМ
.
Освітній рівень: «магістр за ОПП»
.
Спеціальність: 126 «Інформаційні системи та технології»
.

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Терентьєв О.О.
_____
„___” _____2024 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

_____Солодей Нікіта Іванович_____
1.Тема роботи: Моделювання кіберзагроз при впровадженні Інтернету речей
.
затверджена наказом ректора КНУБА № _____ від «____»_____2024 р.
2. Керівник роботи: Горда Олена Володимирівна, к.т.н, доцент кафедри ІТ
_____.
3. Строк подання студентом роботи до захисту: грудень 2024 року
.
4. Зміст пояснювальної записки за розділами:

Р.1. Аналіз предметної області та постановка задачі

.

Р.2. Місце і значення LANDING PAGE в інтернет мережі

.

Р.3. Засоби і методи створення LANDING PAGE

.

Р.4. Розробка програмного забезпечення та тестовий приклад програми

.

Р.5. Ергономіка інформаційних технологій

.

5. Інформаційні слайди:

С.1. Структура сайту типу landing page квитків

.

С.2. Приклад сайту типу landing page

.

С.3. Лідогенеруюча цільова сторінка .

С.4. Можливі інструменти створення Landing page

.

С.5. Набір основних властивостей CSS3, що підтримуються різними браузерами.

С.6. Структура програмного забезпечення

.

С.7. Тестовий приклад програми

.

6. Календарний план виконання атестаційної випускної роботи магістра

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Вересень 2024 р.
Р. 2. Місце і значення LANDING PAGE в інтернет мережі	Вересень 2024 р.
Р. 3. Засоби і методи створення LANDING PAGE	Жовтень 2024 р.
Р. 4. Розробка програмного забезпечення	Жовтень 2024 р.
Р. 5. Ергономіка інформаційних технологій	Листопад 2024 р.
Остаточне оформлення роботи	03 грудня 2024 р.
Направлення роботи на рецензування, перевірку на плагіат	05 грудня 2024 р.

Попередній захист роботи на кафедрі	12 грудня 2024 р.

7. Консультанти розділів атестаційної випускової роботи магістра

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Прийом програмного продукту	к.т.н. доц. Шабала Є.Є.		

8. Дата видачі завдання: 05 вересня 2024 року

Керівник _____ Горда О.В.
(підпис) (прізвище та ініціали)

Магістрант _____ Солодей Н.І.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Солодей Н.І. «Моделювання кіберзагроз при впровадженні Інтернету речей».

Атестаційна випускова робота магістра за спеціальністю: 126 «Інформаційні системи та технології». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Магістерська робота присвячена дослідженню та аналізу потенційних ризиків кіберзагроз, що поширюються через Інтернет речі.

Об’єктом дослідження є процеси та моделі сценаріїв для розповсюдження вірусного програмного забезпечення через Інтернет речі.

Результатом роботи є детальний аналіз поширення конкретного вірусу через Інтернет речі, створення моделі життєвого циклу вірусу, представлені проблеми при впроваджені технології IoT та їх вирішення.

Ключові слова: ІТ, IoT, ВПЗ, ПК, ОС, ПЗ, АП.

SUMMARY

«Simulation of cyber threats in the introduction of the Internet of Things».

Master's attestation final thesis in the specialty: 126 "Information systems and technologies". - Kyiv National University of Construction and Architecture. - Kyiv, 2024.

The master’s thesis is devoted to the study and analysis of potential risks of cyber threats that spread through the Internet of Things.

The object of research is the processes and models of scenarios for the spread of viral software over the Internet of Things.

The result of the work is a detailed analysis of the spread of a particular virus through the Internet of Things, the creation of a model of the life cycle of the virus, the problems presented in the implementation of IoT technologies and their solutions.

Keywords: IT, IoT, VS, PC, OS, SW, HW.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ІТ	інформаційні технології — сукупність методів, виробничих процесів і програмно-технічних засобів, інтегрованих з метою збирання, опрацювання, зберігання, розповсюдження, показу і використання інформації в інтересах її користувачів;
Розповсюджувач	модуль, відповідний за поширення зловмисного коду вірусу через мережу Інтернет речей по вразливим пристроям;
Завантажувач	модуль, відповідний за встановлення шкідливого коду на пристрої;
Звітувач	сканування нових вразливих пристроїв у мережі та звітність;
IoT	технологія пристроїв Інтернет речей;
Вірус	продукт кібератаки або кіберзагрози;
Хакер	людина, яка розробляє та керує вірусним забезпеченням;
Інфікування	результат кібератаки за допомогою шкідливого програмного забезпечення;
ВПЗ	вірусне програмне забезпечення;
ПК	персональний комп’ютер;
ОС	операційна система;
ПЗ	програмне забезпечення;
АП	апаратне забезпечення.

ВСТУП

В наш час Інтернет речі – це наступний крок розвитку Інтернету. Із розвитком різних комп'ютерних технологій почали вдосконалюватися алгоритми аналізу, передачі, збору та розподілу даних. Це дозволяє людству перетворювати різні дані в структуровану інформацію. Тому дане питання являється актуальним в наш час.

Під концепцією IoT розуміють не тільки величезну кількість різнойменних датчиків або приладів, що об'єднані за допомогою безпроводових та проводових технологій і різних протоколів передачі даних через Інтернет, але й взаємодію між віртуальними та реальними світами, в яких діалог відбувається між пристроями та людиною. Якщо більш детальніше, Інтернет речі – це глобальна мережа речей-приладів, які під впливом деякого програмного забезпечення приймають певні рішення з урахуванням тих чи інших чинників, які обробляються без участі людини [1]. Крім звичайних щоденних приладів (холодильник, чайник, годинник, телефон, пилосос тощо) у цій мережі повинні бути залучені різного роду датчики (наприклад, ваги, вібрації, температури і т.п.). Тобто пристрої, що будуть зберігати зібрану інформацію і пристрої, які будуть керувати всіма цими процесами.

В зв'язку з цим ми спостерігаємо про виявлення нових вразливих місць у комп'ютерних мережах та ситемах, серверів різних корпорацій і компаній. Постає питання надійного захисту приватної інформації новими методами. Все більше доводиться чути про відновлення програмного забезпечення чи вимкнення базових функцій пристроїв для забезпечення захисту, знешкодження різної природи комп'ютерних вірусів чи виявлення нових модернізованих.

Як правило, жертвами шкідливого впливу через ресурси мережі Інтернет серед пристроїв зараз може налічувати кожен десятий пристрій. Під шкідливим впливом розуміють наявність стороннього програмного забезпечення, яке має певну дію на стан технологічного продукту або створює нові процеси у операційній системі пристрою.

Прилади-чіпи які виробляються різними відомими компаніями (Cisco, Aruba Networks, Meraki, Intel, AMD тощо) мають вразливі місця у різних технологіях. За допомогою зараження будь-якого пристрою можливо порушити роботу інших пристроїв, що слугують, наприклад, для поширення Інтернет-з'єднання методом встановлення будь-якого програмного забезпечення на пристрої, і за допомогою якого, можливо виводити різні системи з ладу або виконувати на них нештатні команди.

Але важливе те, що від стороннього програмного забезпечення страждають звичайні люди, державні та фінансові установи, а кібертероризм порівнюють зі зброєю масового знищення. Загрози, що несуть злочини в інформаційному просторі, є небезпечними для енергетичних, транспортних та виробничих систем. Тому розуміння процесів, що відбуваються не тільки при запуску шкідливого програмного забезпечення, а і при низькорівневих атак, є необхідним кроком для розробки систем інформаційного захисту. Тому дослідження кіберзагроз з метою визначення шляхів їх поширення є актуальною задачею [2].

Метою роботи є аналіз існуючих кіберзагроз при впровадженні IoT та методи їх вирішення або вдосконалення.

Об'єктом дослідження являються пристрої IoT, мережа фізичних об'єктів, датчиків, сенсорів та відповідних приладів.

За матеріалами магістерської роботи опубліковано наукові роботи, які були представлені на міжнародно науково-практичних конференціях.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення термінів в мережі IoT

IoT (Інтернет речі, Internet of Things) – це сукупність механічних приладів, які мають вбудовані датчики для зчитування і обробки інформації з навколишнього середовища і призначені для передачі її на комп'ютери чи сервери для подальшої обробки. IoT з технологічної точки зору – це, по суті, мережа мереж, що складаються з унікально ідентифікованих об'єктів (по факту "речей"), які можуть взаємодіяти між собою через IP-підключення без втручання людини [3]. Підключені пристрої та машини IoT можуть покращити нашу роботу та життя. Приклади застосування Інтернету речей: від розумного будинку, який автоматично регулює опалення та освітлення, до розумної фабрики, яка контролює промислові машини [4]. На рис. 1.1 зображено пристрої Інтернет речей, що складаються з підмереж таких приладів, що пов'язані між собою однаковим функціоналом чи однією системою керування.

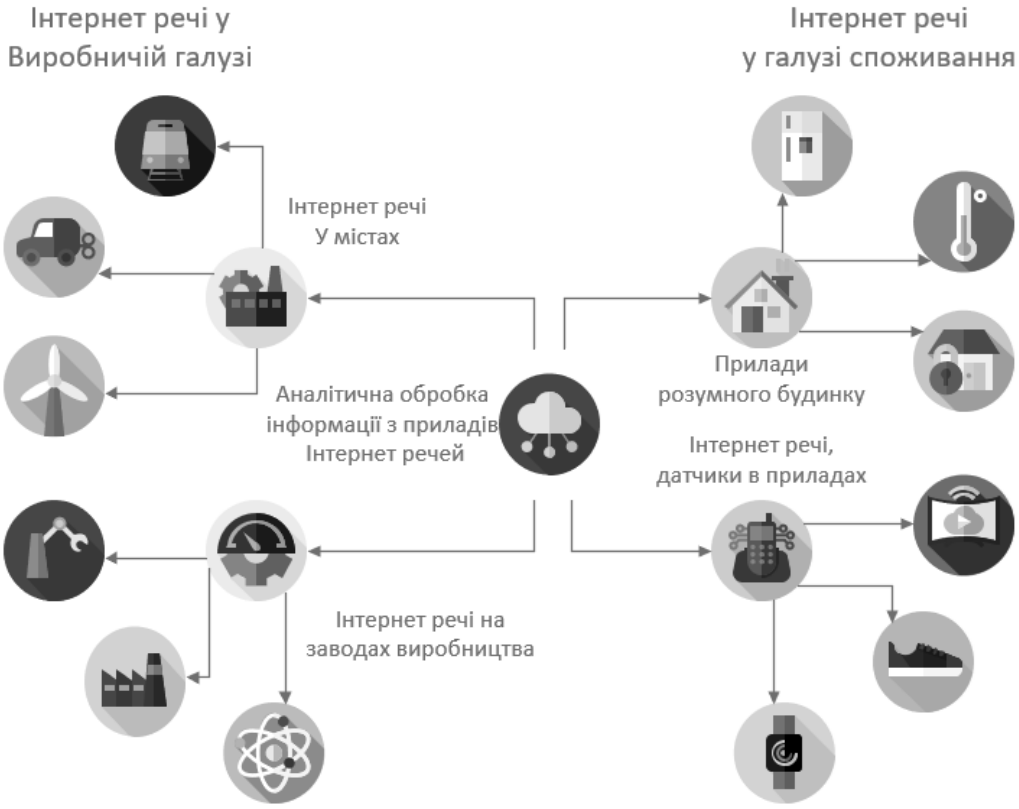

Рис. 1.1 – Структура пристроїв Інтернет речей

Напрямок IoT став активно розвиватися, коли на початку 2000-х років кількість пристроїв, підключених до мережі Інтернет, перевищила кількість користувачів Інтернету. Тобто Інтернет речей перевищив Інтернет людей [5].

Головне завдання IoT – зробити життя людини комфортним та безпечним. IoT супроводжує всі дії користувача та орієнтований на результат [6].

Концепція Інтернету речей є комплексною та передбачає об'єднання таких областей як АЗ, мережі та ПЗ [7].

Інтернет речі ґрунтуються на трьох базових принципах. По-перше, скрізь поширену комунікаційну інфраструктуру. По-друге, глобальну ідентифікація кожного об'єкта. По-третє, можливість кожного об'єкта відправляти і отримувати дані за допомогою персональної мережі або мережі Інтернет, до якої він підключений [8].

Для роботи будь-якої системи IoT потрібне використання ресурсів, апаратного забезпечення, програмного забезпечення та систем. Їх потрібно поєднати в єдине ціле, якими б вони не були різними, щоб сформувати комплексне, надійне та економічно вигідне рішення.

M2M (Міжмашинна взаємодія, Machine-to-Machine) – це клас технологій, які дозволяють пристроям передавати інформацію один одному, не важливо за проводами або бездротовими інтерфейсами [9].

IoT розвивався разом із розвитком M2M. Машини, які з'єднувались один з одним через мережу без взаємодії людини. M2M відноситься до підключення пристрою до хмари, управління ним та збору даних. Піднімаючи M2M на наступний рівень, можна сказати, що IoT – це сенсорна мережа з мільярдів розумних пристроїв, яка з'єднує людей, системи та інші програми для збору та обміну даними. Хоча кожна система IoT різна, основа для кожної архітектури Інтернету речей, а також загальний потік його даних приблизно однаковий. Перш за все, вона складається з "речей", що є об'єктами, які підключені до Інтернету. За допомогою вбудованих датчиків та виконавчих механізмів вони здатні аналізувати навколишнє середовище та збирати інформацію, яка потім передається на шлюзи IoT. Наступний етап складається із систем збору даних

Інтернету речей та шлюзів, які збирають велику масу необроблених даних, перетворюють їх у цифрові потоки, фільтрують та попередньо обробляють, щоб вони були готові до аналізу. Третій етап представлений граничними пристроями, відповідальними за подальшу обробку та розширений аналіз даних. На цьому етапі також можуть бути технології візуалізації та машинного навчання. Після цього дані передаються в центри обробки даних, які можуть бути або хмарними, або встановленими локально [10].

Загальна архітектура Інтернету речей наведена на рис 1.2.

IoT архітектура складається з 4-х частин:

– датчики та приводи;
– шлюзи та збір даних;
– Edge IT (концепція граничних обчислень);
– дата центри (хмара).

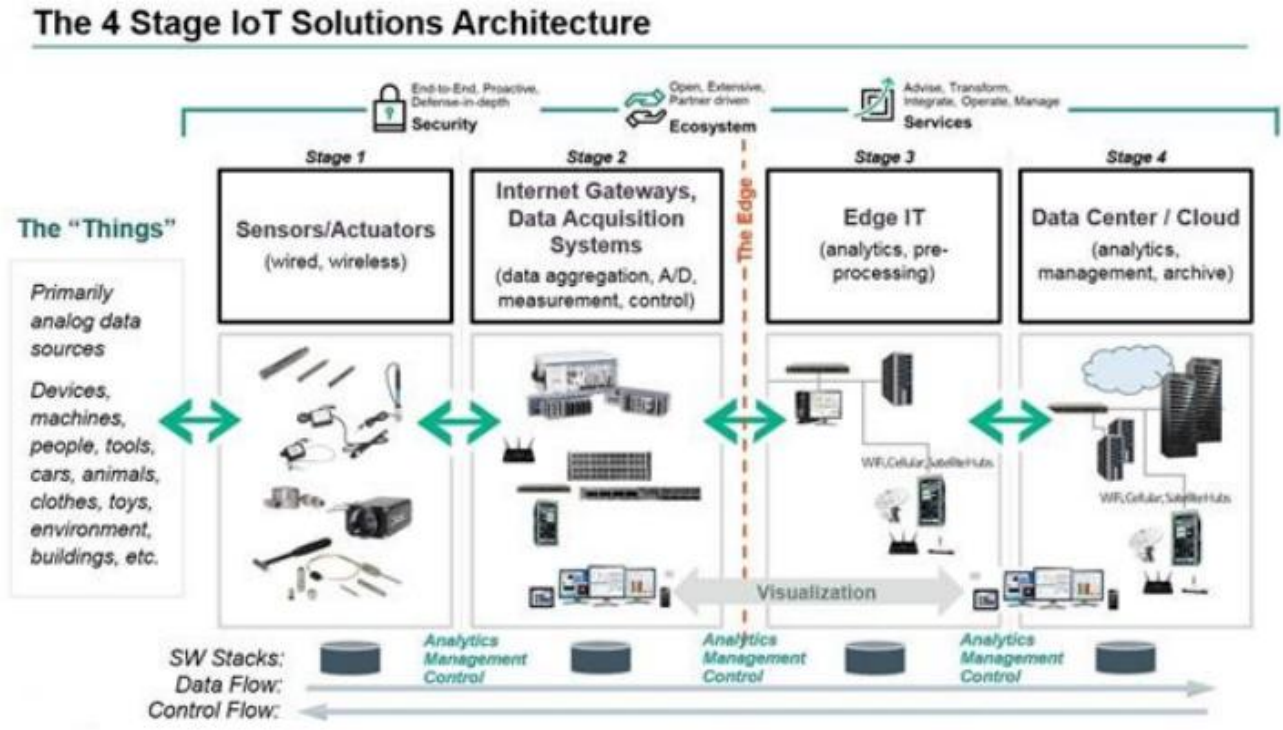

Рис. 1.2 – Основні етапи архітектури Інтернету речей

1.2 Датчики, приводи

Підключені пристрої відповідають за забезпечення сутності Інтернету речей. Щоб зібрати фізичні параметри у зовнішньому світі чи всередині самого об’єкта, їм потрібні датчики. Вони можуть бути вбудовані в самі пристрої або

реалізовані як окремі об’єкти для вимірювання та збору даних телеметрії. Схему наведено на рис. 1.3. Більшість з них потребує підключення через шлюзи датчиків. Підключення датчиків може здійснюватися через локальну мережу (LAN) або персональну мережу (PAN). На основі вхідних даних з датчиків система аналізує ситуацію в режимі реального часу видає спеціальні команди.

Як правило, кожен датчик має свою бібліотеку програм, завантаживши яку на мікроконтролер, ви зможете ним керувати [11].

По суті Інтернет речей – це глобальна мережа комп'ютерів, датчиків (сенсорів) і виконавчих пристроїв (актуаторів), що зв'язуються між собою з використанням інтернет протоколу IP (Internet Protocol) [12].

Отже, надійна архітектура може дозволяти ефективно керувати пристроями лише тоді, коли використовуються потрібні за призначенням безпечні та легкі протоколи зв’язку (наприклад, Lightweight M2M).

Рис. 1.3 – Перша частина IoT архітектури

1.3 Збір даних, шлюзи

Будучи на межі світу експлуатаційних та інформаційних технологій, шлюзи полегшують зв’язок між датчиками та іншою частиною системи, перетворюючи дані сенсорів у формати, які легко передаються та використовуються для інших компонентів системи. Більше того, вони здатні контролювати, фільтрувати та відбирати дані, щоб мінімізувати обсяг інформації, яку потрібно переслати в хмару. Це в свою чергу, позитивно впливає на передачу даних в мережі та час відповіді. Схему наведено на рис. 1.4.

В Інтернеті речей шлюз – це мережеве обладнання, яке виконує безліч функцій, таких як об’єднання десятків датчиків або інтелектуальних пристроїв [13].

Таким чином, шлюзи забезпечують місце для локальної попередньої обробки даних датчиків, які готові для подальшої обробки. Оскільки велика кількість даних отримується за допомогою датчиків, то для передачі даних потрібно високошвидкісні шлюзи і мережі. Ця мережа може мати тип локальної мережі (Wi-Fi, Ethernet і т.п.), широкосмугової мережі (WAN, GSM, 5G і т.п.).

Аспект, який підтримують безпосередньо шлюзи – безпека. Оскільки шлюзи несуть відповідальність за управління потоком інформації в обох напрямках, за допомогою належного інструменту шифрування та захисту вони можуть запобігти витоку хмарних даних IoT, а також зменшити ризик шкідливих зовнішніх атак на пристрої IoT.

Рис. 1.4 – Друга частина IoT архітектури

1.4 Edge IT, концепція граничних обчислень

Edge IT – це апаратні та програмні шлюзи, що аналізують та попередньо обробляють дані перед передачею їх в хмару. Будучи невід’ємною складовою кожної архітектури IoT, граничні пристрої можуть принести значні переваги, особливо великомасштабним проектам IoT. В умовах обмеженої доступності та швидкості передачі даних хмарних платформ IoT граничні системи можуть

забезпечити швидший час реагування та більшу гнучкість в обробці та аналізі даних IoT.

Найважливіша відмінність граничних обчислень від хмарних полягає в тому, що збір та аналіз даних проводиться не в централізованому обчислювальному середовищі, такому як ЦОД, а в тому місці, де відбувається генерація потоків даних. Джерелами даних служать цифрові пристрої (не обов'язково що знаходяться в одній локації), які потім передають ці дані в режимі реального часу (залежить від ситуації, передача інформації може бути відкладена) центральний репозиторій [14].

Оскільки гранична інфраструктура може бути розташована ближче до джерела даних у фізичному відношенні, то IT-служба може простіше та швидше працювати з матеріалам IoT в режимі реального часу та забезпечувати результат у вигляді миттєвої інформації. У цьому випадку сюди пересилаються лише більші фрагменти даних, які дійсно потребують потужності хмари для обробки. За рахунок мінімізації впливу на мережу безпека може бути значно підвищена, тоді як зменшення споживання електроенергії та пропускної здатності сприяє більш ефективному використанню бізнес-ресурсів. Схему наведено на рис. 1.5.

Рис. 1.5 – Третя частина IoT архітектури

1.5 Дата центри, хмарне сховище

Дата-центр (англ. data center) – комплекс для обробки та зберігання даних, що включає приміщення зі стійками і серверами (потужними комп’ютерами), системи клімат контролю, безперебійного живлення та забезпечення безпеки [16].

Якщо датчики є “нейронами”, а шлюз є основою IoT, то хмара – це “мозок” в тілі Інтернету речей і належить до служб управління. На відміну від граничних

рішень, центр обробки даних або хмарна система призначена для зберігання, обробки та аналізу величезних обсягів даних для більш глибокого розуміння з використанням потужних механізмів аналізу даних і механізмів машинного навчання. Схему наведено на рис. 1.6.

Наявні відповідні рішення для користувацьких додатків, хмара може надавати бізнес-аналіз та варіанти презентації, які допомагають людям взаємодіяти із системою, контролювати та відстежувати її та приймати обґрунтовані рішення на основі звітів, інформаційних панелей та даних, що переглядаються в режимі реального часу.

Шість способів використовувати хмарне сховище [17]:

Мобільний робочий простір. Хмара надає будь-якій організації необхідні інструменти для роботи з будь-якого розташування. Електронні листи та всі важливі документи можна передавати до хмари та зберігати там, тому хмара ідеально підходить для роботи. Обмін файлами – це ще одна вагома причина, завдяки якій хмарні технології дають змогу працювати з мобільних пристроїв. Коли ваша організація використовує розміщене хмарне середовище, ви можете передавати туди великі або конфіденційні файли, які не хотіли б пересилати електронною поштою. Деякі постачальники хмарних сховищ дають змогу легко редагувати документи в групах (навіть одночасно!) за допомогою звичних засобів.

Електронна пошта. Розміщення електронної пошти в хмарі зручне тим, що вона буде доступна звідусіль. Це дуже важливо, якщо ваші працівники багато подорожують або працюють віддалено. У сучасному корпоративному світі електронна пошта вважається критично важливим компонентом роботи, тому вона має бути надійною. Якщо ви використовуєте хмару (а не власний сервер для обміну поштою), можете не хвилюватися про його роботу та місткість.

Резервування файлів. Дуже важливо, щоб організації зберігали копії своїх файлів, зокрема тому, що ніхто не знає, які саме документи, електронні листи та інші файли виявляться дійсно цінними. Старі системи для резервування даних використовували жорсткі диски або стрічку. Ці фізичні носії доводилося пересилати до сховищ, як тільки вони заповнювалися. Хмарне сховище для

резервних копій не вимагає таких зусиль і зручне у використанні. Це зокрема стає в пригоді, якщо ви переносите свій офіс або переходите на інше обладнання. Також слід врахувати захист інформації від природніх і антропогенних надзвичайних ситуацій, наприклад пожеж або повеней.

Хостинг веб-сайтів і електронна комерція. У сучасному динамічному світі нікому не подобаються веб-сайти, які завантажуються повільно. Розмістивши свій веб-сайт та інтернет-магазин у хмарі, ви помітите, що час простоювання впаде до практично не помітних значень. До того ж хмару можна масштабувати, зокрема якщо у вас є надійний постачальник корпоративного хмарного сховища. У такий спосіб ви можете легко розгортати додаткові ресурси в дні пікового навантаження, наприклад на свята або під час розпродажів.

Тестування та розробка. Якщо вашій IT-команді потрібне середовище для тестування нової лінійки продуктів або програми, ви можете істотно заощадити час і зусилля, скориставшись хмарним середовищем. У такому середовищі ваша команда може тестувати час завантаження та моделювати поведінку програми в реальному часі, не розгортаючи її в робочому середовищі та не утримуючи окреме тестове середовище.

Бізнес-програми. У хмарі можна розмістити ваші улюблені бізнес-програми, щоб працівники могли використовувати їх будь-де та будь-коли. Вибираючи постачальника хмарного сховища, обов’язково вибирайте таке, яке підтримує безліч бізнес-програм для керування зв’язками з клієнтами, логістики, планування, аналітики тощо.

Рис. 1.6 – Четверта частина IoT архітектури

2 ДОСЛІДЖЕННЯ ПРИНЦИПІВ РОБОТИ ТА ЗАГРОЗ IoT

2.1 Архітектура IoT

Архітектура IoT пристроїв складається з чотирьох основних рівнів [18]:

- рівень датчиків;
- мережевий рівень;
- рівень обробки даних;
- прикладний рівень.

Рівні мережі зображені на рис. 2.1.

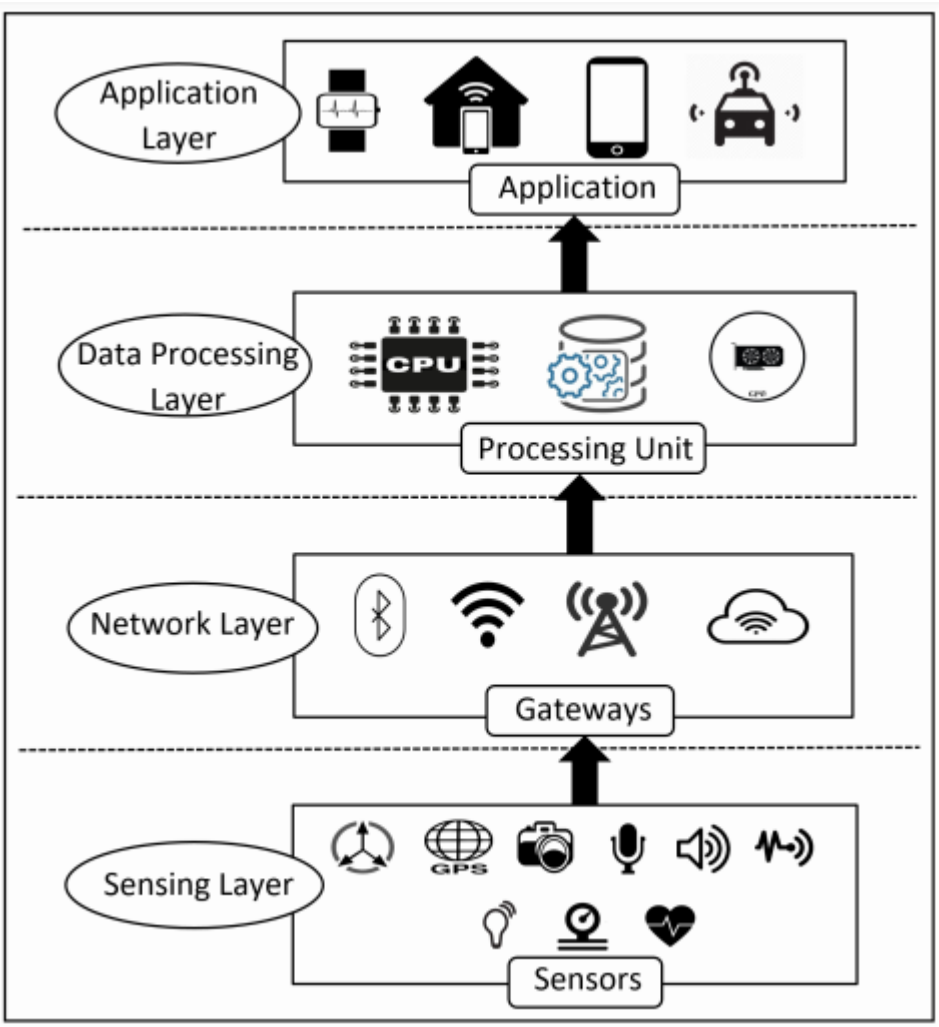

Рис. 2.1 – Основні рівні архітектури мережі

IoT складається з набору різноманітних інфокомунікаційних технологій, що забезпечують його роботу. Архітектура Інтернету речей показує, як різні технології пов’язані між собою [19].

Рівень датчиків – це ідентифікація різних подій навколишнього середовища за допомогою периферійних пристроїв і отримання даних з реального світу [20]. Він складається з декількох видів датчиків. Використання великої кількості цих датчиків є однією з найважливіших функцій IoT пристроїв. Вони зазвичай інтегруються за допомогою концентратора датчиків. В свою чергу, концентратор датчиків є головною точкою з'єднання для кількох інших датчиків. Його функція полягає в акумулюванні та передаванні даних з датчиків до блоку обробки даних в пристрої.

Концентратор датчиків використовує декілька транспортних механізмів таких, як Inter-Integrated Circuit (I2C) чи Serial Peripheral Interface (SPI) для передачі даних між датчиками та додатками.

Датчики в IoT пристроях можуть бути класифіковані на 3 широкі категорії:

- датчики руху;
- датчики навколишнього середовища;
- датчики місцеперебування пристроїв IoT.

Датчики руху вимірюють зміни в русі, а також орієнтацію пристроїв. Є два типи рухів, які можна спостерігати за допомогою датчику цього пристрою: лінійний та кутовий рух. Лінійний рух відноситься до лінійного переміщення пристрою IoT, в той час як кутовий рух відноситься до обертального переміщення пристрою.

До датчиків навколишнього середовища відносяться пристрої, серед яких датчики світла, тиску тощо, що вбудовані в IoT пристрої та реагують на зміни в параметрах навколишнього середовища за допомогою периферійних пристроїв. Основною метою використання цих датчиків в IoT пристроях є допомога пристроям приймати автономні рішення відповідно до змін в периферійних пристроях. Як приклад, датчики навколишнього середовища використовуються в більшості додатків для спрощення життя користувачів (розумні замки, система домашньої автоматизації, розумне освітлення тощо).

Датчики місцеперебування пристроїв IoT взаємодіють з фізичним місцеперебуванням та розташуванням самого пристрою. Найбільш поширеними

датчиками місцеперебування, що використовуються в IoT є магнітні датчики та Global Position System (GPS) датчики. Магнітні датчики використовуються, як цифрові компаси та допомагають фіксувати орієнтацію дисплею пристрою. GPS датчики використовуються для навігаційних цілей в IoT пристроях.

Мережевий рівень використовується, як комунікаційний канал для передачі даних, зібраних на рівні датчиків, до інших підключених пристроїв. В IoT пристроях, мережевий рівень реалізований для використання різноманітних комунікаційних технологій (Z-Wave, LoRa, Wi-Fi, Bluetooth, NFC тощо), що дозволяють передавати дані між різними пристроями всередині самої мережі.

Рівень обробки даних складається з головного блоку обробки даних IoT пристрою. Цей рівень отримує зібрані дані на рівні датчиків та аналізує їх. А вже потім приймає рішення, що базуються на результаті аналізу. В деяких IoT пристроях (смарт-годинник, розумний домашній концентратор і т.п.) цей рівень також зберігає результати попередніх аналізованих даних для подальшої можливості використання цих даних. Також цей рівень може передавати результати обробки даних з одного підключеного пристрою на інший пристрій через мережевий рівень.

Рівень додатків показує результати роботи рівню обробки даних за допомогою різного роду додатків IoT пристроїв. Цей рівень є орієнтованим на користувачів, який виконує різні завдання для користувача. Існують різні IoT додатки, які включають в собі розумний будинок, розумний транспорт тощо.

2.2 Принцип роботи IoT

Зазвичай IoT-система складається з пристроїв із датчиками та хмарної платформи, до якої вони підключені через WiFi, Bluetooth, LAN. Пристрої збирають дані та відправляють їх у хмару, а програма їх обробляє. Програми Інтернету речей дозволяють візуалізувати дані, стежити за показниками та керувати підключеними пристроями [21].

Основні принципи роботи IoT об’єднує чотири різні етапи [22]:

- зчитування інформації за допомогою датчиків;
- передача даних до хмарних сховищ;

– обробка даних, отриманих за допомогою датчиків;
– передача інформації на інтерфейс користувача.

На рис. 2.2 наведено загальний принцип роботи мережі IoT.

Пристрої збирають дані зі свого оточення. Це може бути як простим процесом, таким як зчитування температури, або складним, як запис відео на камеру відеоспостереження. Декілька датчиків можуть бути об'єднані разом чи можуть бути частиною пристрою (рис. 2.2). Наприклад, смартфон – пристрій з кількома датчиками (камера, акселерометр, GPS і т.п.), але смартфон не є лише датчиком.

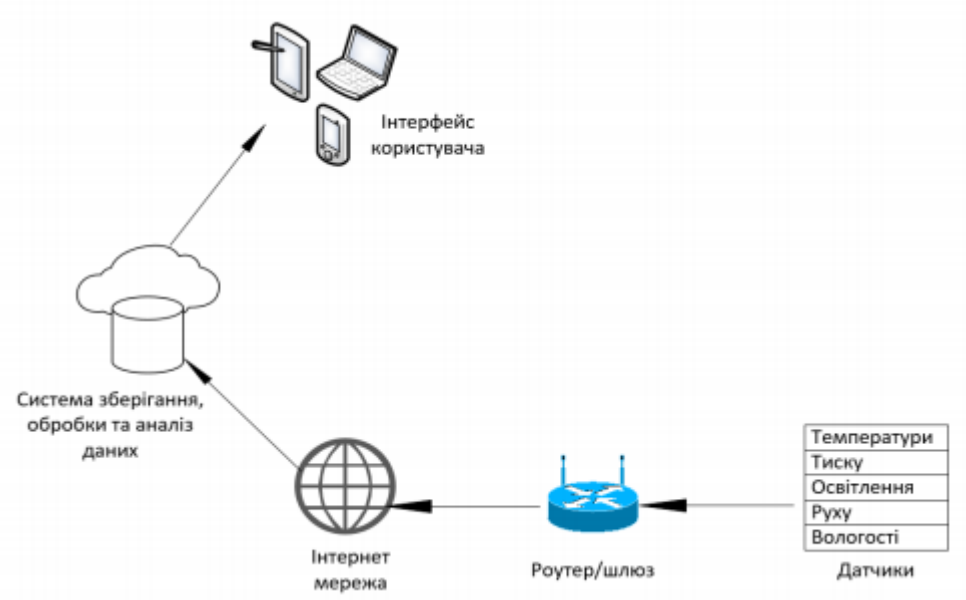

Рис. 2.2 – Схема роботи IoT

Датчики можуть бути підключені до хмари за допомогою різних методів, включаючи: стільникову мережу, супутникову мережу, Wi-Fi, Bluetooth, NFC, малопотужні широкосмугові мережі (LPWAN) або з'єднатися безпосередньо до Інтернету через технологію Ethernet. Кожна опція має компроміс між споживанням енергії, діапазоном та пропускною здатністю. Вибір того чи іншого варіанту підключення зводиться до вибору конкретної області, де буде використовуватися Інтернет речі, але всі вони виконують одне і те ж завдання – передачу даних до хмари.

Після того, як дані потраплять до хмари, ПЗ виконує певну обробку даних. Наприклад, перевірка того, що зчитувана температура знаходиться в межах

допустимого діапазону. Це може бути процесом обробки інформації, наприклад, використання комп'ютерного програмного забезпечення, яке здійснює ідентифікацію об'єктів, що знаходяться на відео (наприклад, зловмисники будівлі).

На етапі передачі інформації на інтерфейс користувача відбувається передача важливої інформації кінцевому користувачу. Це може бути через сповіщення, наприклад, електронна пошта і т.п. Може бути текстове сповіщення, коли температура занадто висока в холодному сховищі будівлі. Крім того, користувач може мати інтерфейс, який дозволяє йому активно перевіряти систему. Наприклад, користувач може перевірити камери відеоспостереження будівлі через телефонну програму або веб-сторінку. Але це не єдиний шлях. Залежно від програми IoT, користувач може також виконувати дії на відстані та впливати на систему. Наприклад, користувач може дистанційно регулювати температуру в холодному сховищі за допомогою спеціального ПЗ на смартфоні. А ніші дії можуть виконуватися автоматично. Замість того, щоб чекати, коли користувач налаштує температуру, система може зробити це автоматично за допомогою попередньо визначених правил. Як приклад, замість того, щоб викликати вас до будинку, те стався грабіж, система IoT також може автоматично повідомляти про це відповідні органи. Система IoT складається з датчиків, які "спілкуються" з хмарою через певний вид зв'язку. Відразу дані, які потрапляють у хмару, ПЗ обробляє їх, а далі може приймати певне рішення, наприклад, надіслати попередження чи автоматично регулювати роботу датчика без потреби користувача.

Можливості управління IoT охоплює традиційні класи несправності, конфігурації обліку, показників роботи та безпеки (FCAPS). Тобто управління несправностями, управління конфігурації, управління обліком, управління показниками роботи та управління безпекою.

Найважливіші загальні можливості управління в IoT включає:

– управління пристроями (наприклад, дистанційне ввімкнення або вимкнення пристроїв, їх діагностика, оновлення прошивки та/або програмного забезпечення, керування робочим станом пристроїв);
– управління топологією локальної мережі;
– управління трафіком і перевантаженнями (наприклад, виявлення умов перевантаженості мережі та реалізації резервування ресурсів для термінових або життєво важливих потоків трафіку).

Спеціалізовані можливості управління тісно пов’язана з вимогою додатків. Як приклад, вимогами по контролю лінії передачі електроенергії в розумній електромережі.

Існує два види можливостей забезпечення безпеки: загальні та спеціалізовані. Загальні можливості забезпечення безпеки не залежить від додатків і включає:

1) на прикладному рівні:
– авторизацію;
– автентифікацію;
– Захист конфіденційності та цілісності даних додатків;
– захист недоторканності приватного життя;
– аудит безпеки;
– антивірусне ПЗ;
2) на мережевому рівні:
– авторизацію;
– автентифікацію;
– конфіденційність даних про використання та даних сигналізації;
– захист цілісності даних сигналізації;
3) на рівні пристроїв:
– авторизацію;
– автентифікацію;
– перевірку цілісності пристрою;
– управління доступом;

– захист конфіденційності;
– цілісність даних.

Спеціалізовані можливості забезпечення безпеки взаємопов'язані з вимогами додатків, наприклад, вимогами безпеки мобільних платежів.

2.3 Протоколи IoT

Протоколи Інтернет речей є важливою частиною стека технології, без них обладнання буде безкорисним, оскільки протоколи дозволяють IT-підрозділу обмінюватися даними. З цих переданих фрагментів даних можна витягти корисну інформацію для кінцевого користувача і завдяки цьому розгортання мережі стає економічно вигідним.

Існують різні підходи до класифікації протоколів Інтернету:

– D2D (пристрій до пристрою) – протокольні взаємодії навколишніх пристроїв між собою;
– D2S (пристрій до серверу) – для передачі даних, зібраних пристроями на сервери для обробки;
– S2S (сервер до сервера) – протоколи взаємодії серверів один з одним.

Інший має таку структуру: замість того, щоб намагатися встановити всі протоколи IoT поверх існуючих моделей архітектури, таких як OSI, було розбито протоколи на наступні групи, щоб забезпечити певний рівень організації:

– інфраструктурна (6LowPAN, IPv4/IPv6, RPL);
– ідентифікація (EPC, uCode, IPv6, URI);
– комунікації/транспорт (Wifi, Bluetooth, LPWAN);
– відкриття (Physical Web, mDNS, DNS-SD);
– протоколи даних (MQTT, CoAP, AMQP, Websocket, Node);
– управління пристроями (TR-069, OMA-DM);
– семантичний (JSON-LD, Web Thing Model);
– багаторівневі фрейморки (Alljoyn, IoTivity, Weave, Homekit).

Нижче буде опис протоколів даних для IoT.

MQTT – це протокол обміну повідомленнями з шаблоном публікації-підписки (pub/sub) і призначений для зв'язку комп'ютеризованих пристроїв,

підключених між собою до локальної або глобальної мережі, різними громадськими чи приватними веб-сервісами. Його основна мета – заміна приватних технологій, що використовуються різними компаніями і стати стандартом обміну даними в мережі Інтернет, як протокол HTTP [23]. Протокол MQTT спочатку був створений для датчиків, які відстежують стан труб, проте пізніше сфера його діяльності була розширена і він знайшов своє застосування у безлічі вбудованих рішень, в тому числі в смартфонах. Так соціальна мережа Facebook застосовує цей протокол для обміну повідомленнями (Facebook Messenger). Схему протоколу наведено на рис. 2.3.

У мережі на базі протоколу MQTT розрізняють 3 об’єкти:

1) видавець (publisher) – MQTT-клієнт, який при виникненні певних подій передає інформацію про них в брокер;
2) брокер (broker) – MQTT-сервер, який приймає інформацію від видавців і передає її відповідним підписникам;
3) підписник (subscriber) – MQTT-клієнт, який після підписки у відповідного брокера більшу частину часу “слухає” його і постійно готовий до приймання та обробки вхідного повідомлення від брокера.

Протокол знайшов широке застосування в таких пристроях IoT, як електролічильники, транспортні засоби, сповіщувачі, промислове чи санітарне обладнання.

MQTT добре відповідає наступним потребам:

– мінімальне використання пропускної здатності;
– робота з бездротовими мережами;
– низьке споживання енергії;
– висока надійність при необхідності;
– використовує мало ресурсів для обробки та пам’яті.

Незважаючи на свої характеристики, MQTT може бути проблематичним для деяких дуже обмежуючих пристроїв, через факт передачі повідомлень за допомогою TCP та керування назви довгих тем. Це вирішено за допомогою варіанту MQTT-SN, який використовує UDP та підтримує індексацію назв теми.

Однак, незважаючи на широке прийняття, MQTT не підтримує чітко визначену модель представлення даних та структуру управління пристроями, що робить реалізацію своїх можливостей управління даними та можливостями управління пристроями повністю орієнтованими на платформу чи постачальника.

CoAP (протокол обмеженого застосування) – розроблений для вирішення завдань взаємодії пристроїв з обмеженими ресурсами. Практично всі кінцеві пристрої Інтернету речі можна віднести до цього визначення. І саме такі пристрої виконують основну роботу. Концепція CoAP значно відрізняється від MQTT і орієнтується на взаємодію «клієнт-сервер». Клієнт звертається до серверу і відправляє найпростіші команди типу PUT, POST, GET, DELETE, сенс яких

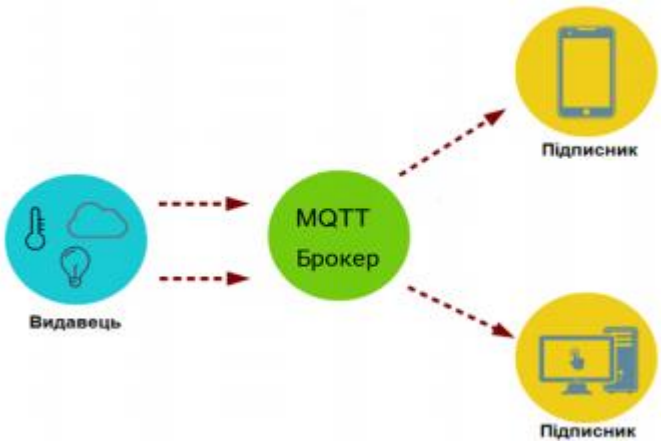

Рис. 2.3 – Схематичне зображення принципу дії протоколу MQTT

зрозумілий з назви та аналогічний HTTP. З цієї точки зору можна сказати, що COAP – це такий спрощений HTTP, в якому мало ресурсів. В результаті варто відмітити легку та просту інтеграцію CoAP з HTTP. За допомогою звичайного браузеру користувач може інтегрувати систему управління пристроями Інтернету речей в звичайний веб-додаток та не помічати цього, а в деяких випадках навіть можна і не здогадуватись цього. З цієї точки зору протокол відповідає вимогам RESTfull. Схему протоколу наведено на рис. 2.4.

CoAP покладається на протокол User Datagram (UDP) для встановлення безпечного зв’язку між кінцевими точками. Використовуючи широкомовну передачу та багатоадресну передачу, UDP здатний передавати дані багатьом хостам, зберігаючи швидкість зв’язку та низьке використання пропускної здатності, що робить його відмінною підсистемою для бездротових мереж,

зазвичай використовуваних у середовищі M2M з обмеженими ресурсами. CoAP має якість обслуговування, яка використовується для контролю надсилаються повідомлень і позначає їх як «Підтверджені» або «Непідтверджені». Це вказує, чи повинен одержувач повернути «Підтвердження» чи ні.

Інші цікаві особливості CoAP полягають у тому, що він підтримує механізм узгодження вмісту та механізм виявлення ресурсів. Крім передачі даних IoT, CoAP використовує безпеку транспортного рівня Datagram Transport Layer Security (DTLS) для безпечного обміну повідомленнями на транспортному рівні.

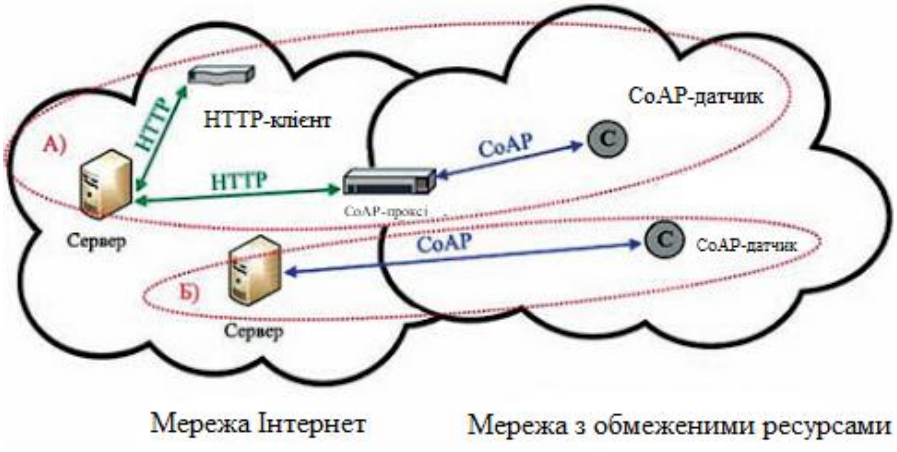

Рис. 2.4 – Схематичне зображення принципу дії протоколу CoAP

CoAP повністю відповідає потребам надзвичайно легкого протоколу, щоб задовольнити потреби пристроїв, які працюють на акумуляторних батареях або з низьким енергоспоживанням.

XMPP (розширюваний протокол обміну повідомленнями та присутності) – призначений для обміну повідомленнями в реальному часі. Цей протокол IoT призначений для комунікації проміжного програмного забезпечення, яке орієнтоване на повідомлення, що засновані на мові XML. Це дозволяє в режимі реального часу обмінюватися структурованими, але розширюваними даними між двома або більше мережевими клієнтами. Схему протоколу наведено на рис. 2.5.

З моменту свого створення XMPP широко застосовується як протокол зв’язку. З часом і з появою легкої специфікації XMPP: XMPP-IoT, вона стала використовуватись в контексті Інтернету речей.

Сильні сторони XMPP – це адресація та можливість масштабування, що робить його ідеальним для розширення IoT, орієнтованого на споживача. Серед недоліків використання XMPP в IoT-зв'язку слід зазначити, що він не пропонує ні якості обслуговування, ні шифрування в кінці. Зважаючи на ці обмеження серед інших, передбачається, що його застосування в IoT залишатиметься слабко пов'язаним, оскільки протокол не став широко застосовуваним для обміну даними та управління пристроями з обмеженими ресурсами, такими як і MQTT або LwM2M.

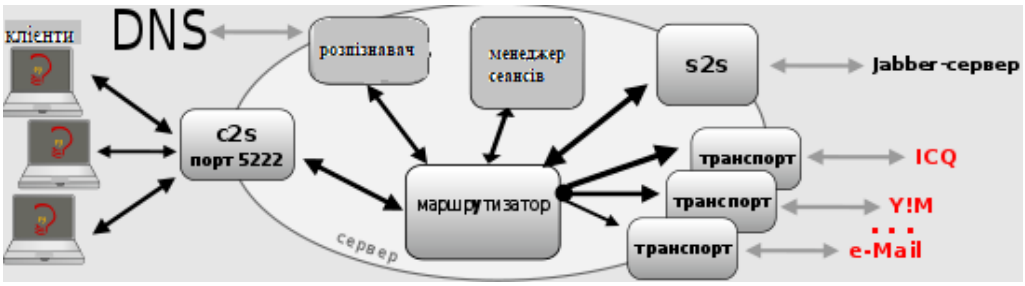

Рис. 2.5 – Схематичне зображення принципу дії протоколу XMPP

M2M (машина-з-машиною) – розроблений по суті для зменшення енергоспоживання та споживання даних. Розроблений для пристроїв малої потужності з обмеженими можливостями обробки та зберігання [15]. Специфікація даного протоколу також описує безліч типових функцій управління пристроєм, такі як дії віддалених пристроїв, оновлення прошивки та програмного забезпечення (FOTA & SOTA), моніторинг і управління підключенням, також включаючи управління стільниковим зв'язком. Схему протоколу наведено на рис. 2.6.

На відміну від будь-якого іншого протоколу IoT на ринку, архітектура LwM2M підтримує чотири логічні інтерфейси, що допомагають стандартизувати спосіб власне управління пристроєм та телеметрією:

Інтерфейс завантаження – дозволяє керувати пристроями без головного пристрою. Це означає, що можна налаштувати пристрій для надання потрібної послуги без необхідності попереднього налаштування на заводі, що значно скорочує витрати та оптимізує час виходу продукту чи послуги на ринок;

Інтерфейс реєстрації клієнта – повідомляє серверу про "існування" клієнта та підтримуваний функціонал. Крім того, він дозволяє оновити прошивку та

програмне забезпечення по хмарі. Інтерфейс управління пристроєм та підтримка сервісу LwM2M дозволяє постачальнику отримати доступ до ресурсів об'єкта, що дозволяє йому змінювати налаштування та параметри пристрою;

Інтерфейс інформаційних звітів – завдяки взаємодії з публікацією/підпискою користувач може отримувати з пристроїв звіти про помилки, коли сервіс не працює належним чином, а також надсилати запити про стан пристрою.

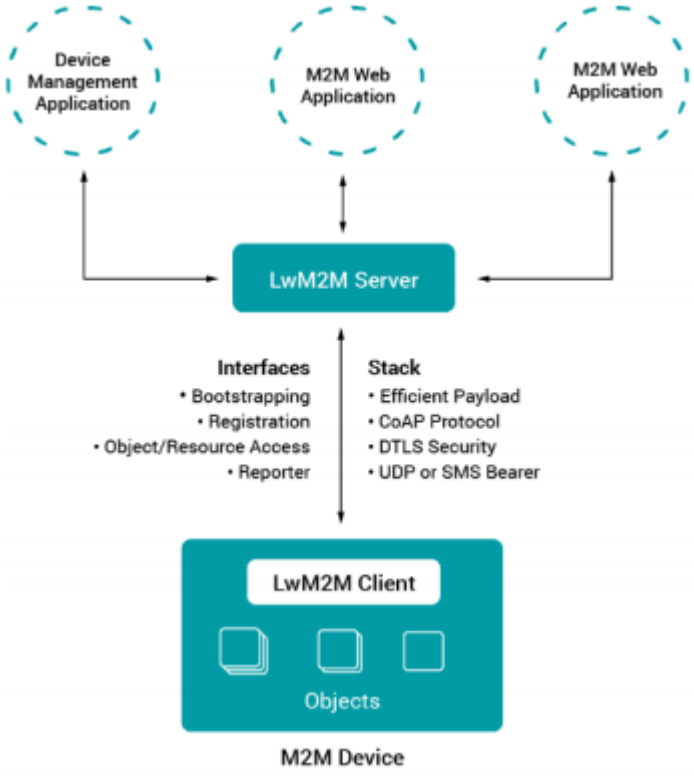

Рис. 2.6 – Схематичне зображення принципу дії протоколу LwM2M

DDS (служба розподілу даних) – протокол, який в режимі реального часу для M2M-зв'язку забезпечує масштабований, надійний, високоефективний та сумісний обмін даними між підключеними пристроями, незалежними від апаратного забезпечення та програмної платформи. DDS підтримує архітектуру без брокерів та багатоадресну передачу для забезпечення високої якості QoS та забезпечення сумісності. Архітектура протоколу DDS базується на рівні Data Centric Publish-Subscribe (DCPS) та рівні Data-Local Reconstruction Layer (DLRL). В той час як DCPS відповідає за масштабованість та розподіл даних з урахуванням ресурсів, DLRL пропонує інтерфейс для функціональних можливостей DCPS, що

дозволяє передавати дані між об’єктами, підключеними до IoT. Не дивлячись на те, що DDS є не типовим рішенням для IoT, він все ще знаходить своє застосування в деяких розгортаннях індустріального Інтернету речей, таких як: управління повітряним рухом, управління розумними мережами, автономні транспортні засоби, транспортні системи, робототехніка, виробництво електроенергії та послуги охорони здоров’я. В цілому DDS може використовуватися для управління обміном даними між легкими пристроями та взаємозв’язку великих, високоефективних сенсорних мереж. Він також може надсилати та отримувати дані з хмари. Схему протоколу наведено на рис. 2.7.

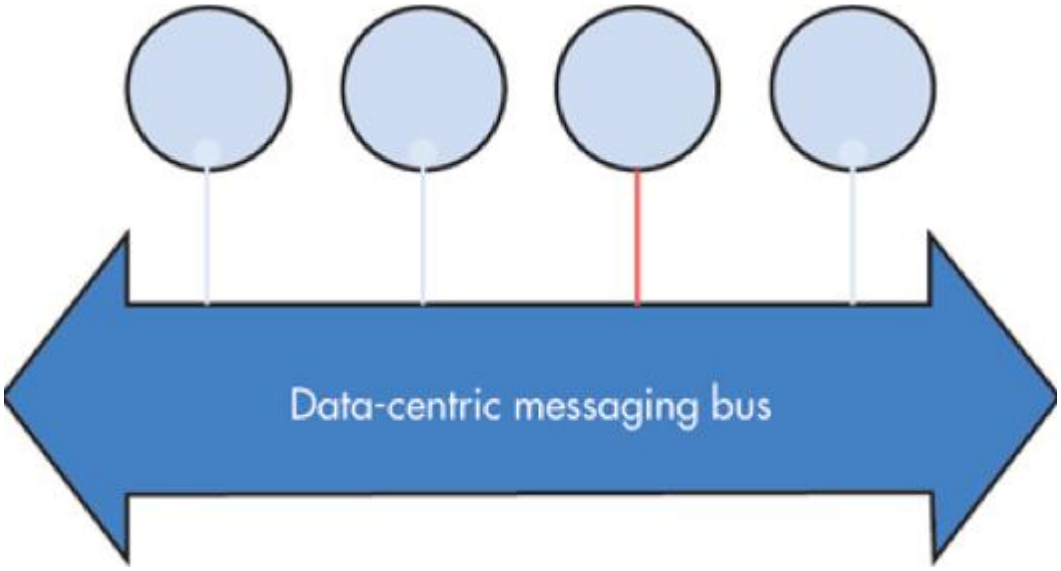

Рис. 2.7 – Схематичне зображення принципу дії протоколу DDC

2.4 Визначення проблем із використанням пристроїв IoT

Виявлення та протидія кіберзагрозі в сучасних інформаційних системах спираються на вже відомі загрози. Але кіберзлочинці постійно знаходять нові інструменти та розробляють все вдосконаліші програмні забезпечення, яким не можуть протидіяти системи інформаційного захисту. Систематизація та узагальнення інформації про шкідливе ПЗ, визначення початкової і кінцевої мети, способи їх використання в кібератаках є важливим ресурсом для побудови сценаріїв кіберзагроз які, частково зможуть описати майбутні атаки і посприяють розробкам нових систем інформаційного захисту.

2.5 Методики вирішення проблем кіберзагроз на пристроях IoT

Дослідження систем і методів захисту пристроїв Інтернет речей можна поділити на два підходи: теоретична і практична. Теоретичну частина описує можливі технології втручання та порушення роботи пристроїв Інтернет речей. Практичну частина описує дослідження існуючих вразливих місць і тих випадків, що вже слугують методами доступу до контролю та інформації пристроїв. Дослідження можна робити опираючись на те, коли пристрої вже стали цілями для враження або в макромасштабі, прогнозуючи майбутні наслідки і можливі атаки у майбутньому.

Варто розуміти, що кожен пристрій Інтернет речей не може бути повністю захищений від будь-яких вразливостей. Як показує практика, для того, щоб система працювала стабільно, потрібно жертвувати іншими характеристиками. У даному випадку зручністю таких пристроїв є їхня мобільність, технологічність і простота у використанні. Для отримання таких критеріїв виробники пожертвували захистом.

Вразливість пристроїв інтернет речей поділяється на два типи:

- взаємодія пристроїв між собою через технології передачі даних;
- апаратне зараження.

Якщо зараз всі дослідження направлені на доопрацювання безпеки протоколів технології передачі даних, то на такі чинники, як захист інфраструктури життєзабезпечення функціонування приладів, не завжди звертають увагу. З великою долею ймовірності можна стверджувати, що при встановленні розумних датчиків нехтують перевірянням станів і захищеності ліній електропередач. В таких випадках потрібно завжди перевіряти наявність резервного кабелю підключення, який при виході з ладу основного, зможе взяти підтримку функціонування пристрою на себе.

Однією з найрозповсюдженіших методів отримання доступу через механізми передачі даних є втручання у бездротову мережу. Це засвідчує про вразливості у інтерфейсі IEEE 802.11, а також наявність внутрішніх атак. Вразливості через аутентифікацію, ідентифікацію та шифрування не завжди є

ключовими цілями для проникнення. Внутрішні атаки – це насамперед прилади або чинники, що викликають вразливості у мережі всередині самої системи захисту. У ролі такого “збудника” внутрішньої атаки може бути будь-що, що має доступ до бездротової чи дротової мережі.

Для реалізації своїх цих цілей кіберзлочинці використовують фізичні вразливості пристроїв або спеціальне програмне забезпечення. Для доступу до особистих інформаційних ресурсів жертви злочинці використовують складності реалізацій технологій, фізичні пристрої, що порушують зв’язок, або створюють різного роду кібератаки чи комп’ютерні віруси, які здійснюють копіювання, модифікацію або знищення інформації.

Основними цілями кібератак є:

- отримання доступу до державних чи військових секретів, а також банківської та особистої інформації (як приклад, вірус Red October);
- нанесення збитків окремим фізичним елементам інформаційного середовища (як приклад, мережевий хробак, ботнет Mirai);
- крадіжка або знищення інформації методами обходу систем захисту та впровадження стороннього програмного забезпечення (як приклад, вірус Wanna Cry);
- захоплення каналів розповсюдження інформації;
- створення перебоїв у роботі мережевих комунікацій.

Всі бездротові мережі в будь-якому випадку з’єднуються з дротовою мережею. Відповідно, при проникненні у дротову мережу чи навпаки, зараження розповсюджується по всім пристроям.

Вірус може проходити через програмно-апаратні методи захисту локальних мереж, такі як:

- мережеві екрани;
- криптографічний захист;
- антивірусні програми;
- біометричні технології аутентифікації;
- віртуальні захищені канали.

Для передбачення цих дій необхідні спеціальні методи, що можуть аналізувати попередні дії зловмисника та, на їх основі, визначати кінцеву ціль атаки і наступні його кроки.

2.6 Запобігання кіберзагрозам пристроїв IoT

Метою будь-якого шкідливого ПЗ, таких як троянські коні, спам-боти, мережеві хробаки і т.п., є зараження пристрою і виконання протиправних дій на ньому. Всі вони скоріше за все мають функціонал об'єднання пристроїв у ботнет мережу. Під ботнет мережею можна розуміти сукупність пристроїв, заражених конкретним екземпляром вірусу, що слугують для збереження шкідливого коду та його розповсюдження, збирання конфіденційної чи приватної інформації, порушення звичайної роботи пристрою або виведення його з ладу, а також виконувати масові запити на сервера різних установ. Таке явище має назву DDOS-атака.

DDOS-атака (відмова в обслуговуванні) – це атака серверів або сайтів через використання потужності багатьох пристроїв з надмірною кількістю звичайних запитів (відвідання ресурсу, авторизація на ресурсі, тощо). Метою атаки являється обробка на апаратному рівні великого об'єму вхідної інформації, наслідком якої є вихід з ладу пристроїв. На даний момент діючими програмами слугують атаки, що створюють мережі з пристроями, не притягуючи до себе увагу до того часу, поки не отримають відповідні команди. Такі атаки після зараження імітують звичайні процеси або знаходяться в пасивному стані.

2.7 Переваги IoT

До переваг в IoT відносять:

- Взаємозв'язок: IoT заохочує зв'язок між пристроями (M2M), це дає змогу без втручання людини машинам спілкуватися один з одним, що призводить до швидшого і своєчасного обміну інформацією;
- Легкий доступ: за допомогою ристрою IoT можна з легкістю перевіряти інформацію, наприклад, перевірка стану продуктів в холодильнику тощо [24];

– Інформативність: більше інформації допомагає приймати кращі рішення;
– Моніторинг: знаючи точну кількість запасів або якість повітря в комнаті можна надати додаткову інформацію та підвищити безпеку в цілому;
– Економія часу: можна заощадити велику кількість часу;
– Економія витрат: оптимальне використання енергії та ресурсів може бути досягнуто шляхом використання цієї технології, сповіщення у випадку можливих проблемних місць, поломок чи пошкодження системи [25];
– Економія електроенергії;
– Автоматизація: дозволяє контролювати завдання, які виконуються щодня, уникаючи втручання людини, за допомогою технології M2M, що допомагає підтримувати прозорість у процесах та вжити необхідних заходів у разі виникнення надзвичайних ситуацій;
– Ефективність роботи: взаємодія між машинами забезпечує кращу ефективність, отже точні результати можна отримати швидко, що призводить до економії цінного часу і дає можливість людям робити інші творчі роботи;
– Краща якість життя: всі додатки цієї технології завершуються підвищенням комфорту, зручності та кращого управління, тим самим покращуючи якість життя.
– Високий рівень безпеки користувача [26];
– Переваги безпосередньо для бізнесу: аналітика для виявлення бізнес-ідей та можливостей.

2.8 Недоліки IoT

Виявлено та наведено деякі недоліки IoT:

– Сумісність пристроїв: пристрої різних виробників мають питання сумісності в з’єднанні та моніторингу;
– Складність мережі: IoT є різноманітною і складною мережею, тому будь-які помилки програмного або апаратного забезпечення матимуть серйозні наслідки (наприклад, збій живлення);

– Менша зайнятість людського персоналу: кваліфікований спеціаліст може втратити роботу внаслідок автоматизації щоденної діяльності, що призводить до проблем безробіття в суспільстві, де в свою чергу зменшується попит людських ресурсів;
– Контроль над життям: життя все більше буде контролюватися різними технологіями та буде залежати від нього;
– Конфіденційність: ризик втрати конфіденційності при передачі даних збільшується [36];
– Безпека: побутові прилади, промислове обладнання, послуги державного сектору (водопостачання і транспорт), а також багато інших пристроїв, які підключені до Інтернету, зберігають багато інформації мережі, де вона може підвергатися зміненню, спотворенню або видаленню [38];
– Цілісність сигналів і живлення: різні електромагнітні перешкоди [40].

Також до недоліків можна віднести проблеми із впровадженням IoT-проектів. Зокрема:

– неправильно підібраний кейс для впровадження;
– нестача кваліфікованих кадрів;
– нездатність правильно збирати та інтерпретувати дані [33].

2.9 Визначення основних етапів кіберзагроз

Планування і реалізація кібератаки складається з кількох етапів:

1) збір інформації;
2) інфікування пристрою або мережі;
3) маніпуляція даними та розповсюдження атаки.

На першому етапі, що зображено на рис. 2.8., виконується втручання в мережу методами шкідливого коду, тобто використання вузькоспеціалізованих мов програмування, вразливих місць специфікацій і стандартів.

На другому етапі, що зображено на рис. 2.9., відбувається зараження пристрою, який являється об’єктом кібератаки.

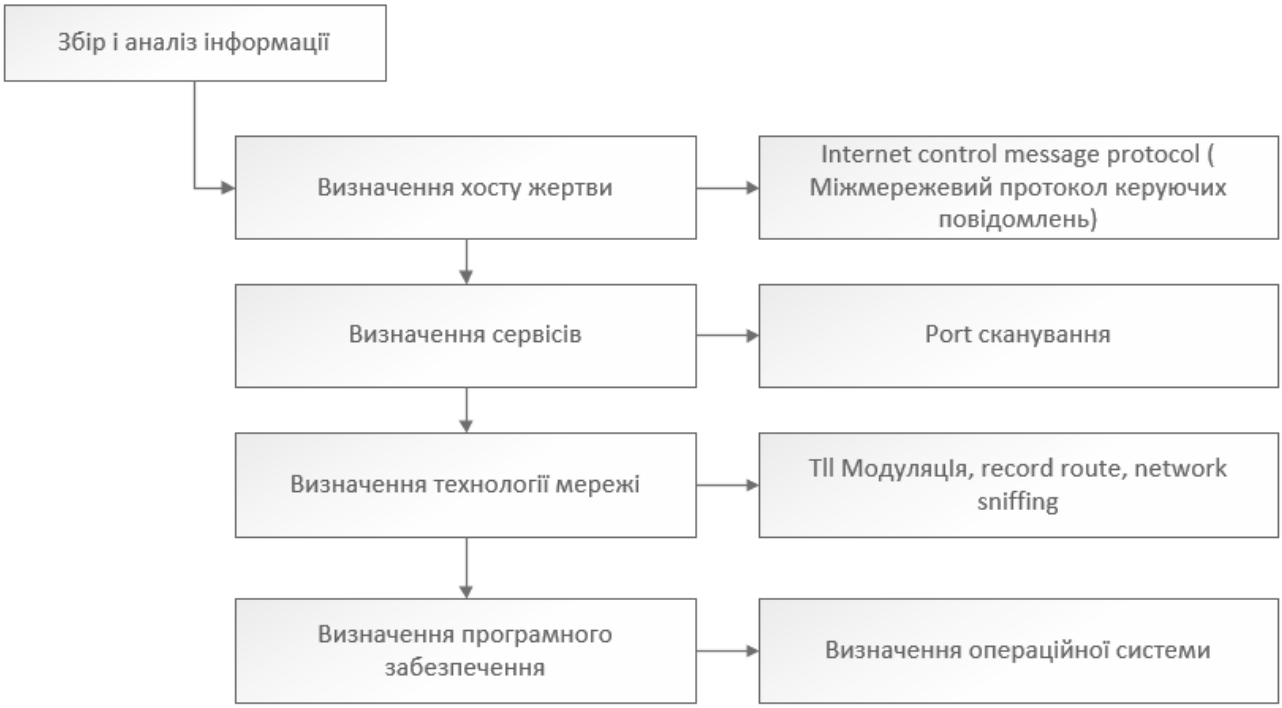

Рис. 2.8 – Збір та аналіз інформації

Після успішного інфікування пристрою відбувається третій етап – маніпуляції з даними та виконання розповсюдження атаки. Його наведено на рис. 2.10.

Рис. 2.9 – Інфікування пристрою або мережі

Рис. 2.10 – Маніпуляція даними та розповсюдження атаки

2.10 Загрози безпеки IoT

Порушення роботи пристроїв інтернет речей фізичним втручанням зазвичай виконуються створенням нестандартних умов для самого приладу. Під такими умовами розуміють порушення роботу приладу методами взаємодії технологіями генерації радіочастотних хвиль (Bluetooth, RFID, NFC) або звичайною підміною ідентифікаторів для зчитування (QR) [28]. На основі імітації загроз та контрзаходів або моделювання результатів кібер-атак, визначається поведінка людей і як вона сприяє на подальші прийняття рішень [22].

Найпоширеніші слабкі місця IoT [27]:

– звичайні облікові записи без врахування перевірки аутентифікації;
– відсутність підтримки з боку виробника для усунення вразливостей;
– складнощі або неможливість своєвчасного оновлення ПЗ чи ОС;
– застосування текстових протоколів та непотрібних відкритих портів;

– використання одного й того ж самого гаджета, через який легко потрапити на всю мережу;
– використання незахищених мобільних технологій;
– використання незахищеної хмарної інфраструктури;
– використання небезпечного ПЗ.

2.11 Загрози безпеки для приватних осіб

Якщо говорити про системи автоматизації для приватних осіб, то з описаного вище найважливішими є питання забезпечення контролю доступу до системи, безпечних комунікацій між її елементами, а також стійкості до атак. При цьому з фінансових причин часто доводиться йти на компроміс між вартістю обладнання та рівнем захисту. Додатково варто зазначити бажаність використання надійних каналів зв'язку та резервних ліній комунікацій, гнучкі можливості системи під час роботи в автономному режимі, а також забезпечення конфіденційності. Ще одним питанням, яке рідко вирішується виробниками масового обладнання, є моніторинг роботи, аналіз та повідомлення про позаштатні ситуації [9].

Конфіденційність особистих даних сьогодні є актуальною темою в багатьох областях. Як приклад, переважна більшість фітнес-трекерів працює спільно з власними хмарними сервісами для зберігання та обробки даних. Як наслідок, користувачам необхідно погоджуватися з багатосторінковими документами умов використання сервісів, якщо вони хочуть скористатися даним пристроєм. Як приклад наведено на рис. 2.11.

Рис. 2.11 – Залежність між сервісами IoT

2.12 Загрози безпеки для організацій

IoT зустрічається не тільки в приватному секторі, а й у багатьох комерційних областях, включаючи транспорт, системи автоматизації, моніторинг навколишнього середовища медицина, тощо. У кожній із них споживачі можуть зустрітися з унікальними загрозами. Наприклад, відстеження транспорту може розкрити бізнес-стратегії компанії, а неправильні показання датчиків можуть призвести до аварії та псування дорогого обладнання. У загальному випадку схема системи з використанням IoT складається з кінцевих пристроїв, опціональних шлюзів, мережі та обчислювальних центрів. Як приклад наведено на рис. 2.12.

Якщо припустити, що в існуючій інфраструктурі серверів та основних комунікацій вже реалізовано всі необхідні технології (зокрема, для корпоративних мереж та баз даних), то для IoT необхідно задуматися про стратегію захисту кінцевих точок, шлюзів та їх комунікацій. Кінцеві точки найчастіше працюють в умовах обмежених ресурсів, включаючи обчислювальні потужності, обсяги пам’яті та енергоспоживання. Для них може бути дуже складно чи навіть неможливо використовувати такі технології, як виділені чіпи безпеки, контроль

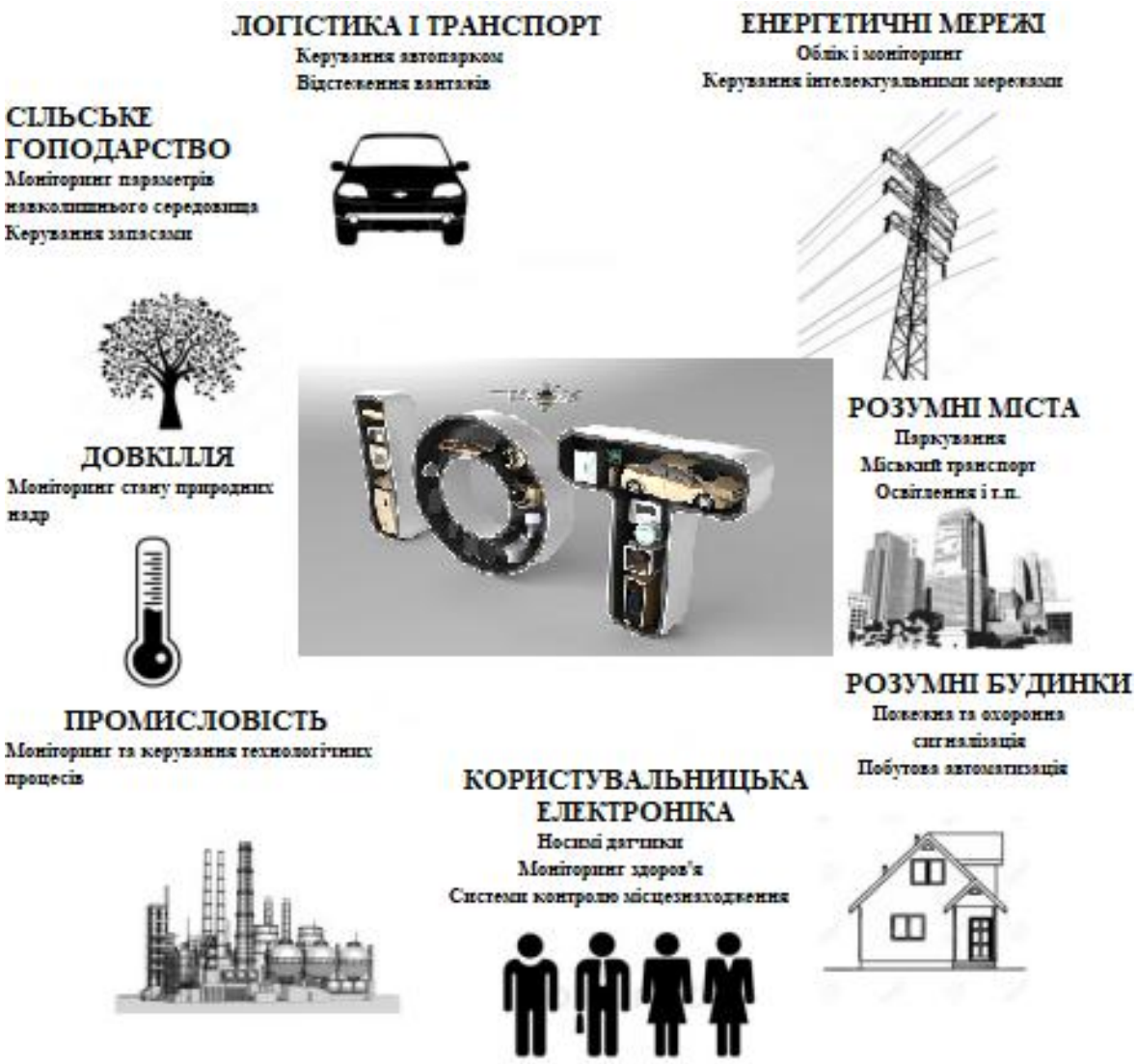

Рис. 2.12 – Технології IoT в комерційних областях

привілеїв, захист пам'яті чи віртуалізацію. Крім того, брак ресурсів може призводити до слабкої стійкості до атак, спрямованих на відмову в обслуговуванні. Питання додатково ускладнюється тим, що все частіше пристрої засновані на власних досить складних мікроконтролерах і підтримують віддалене оновлення вбудованого програмного забезпечення забезпечення. Так що постають завдання щодо забезпечення контролю коректності працюючого на пристроях коду. Як один з варіантів тут пропонуються системи забезпечення довіреного середовища виконання, що базуються на апаратно-програмних рішеннях.

Пропонований деякими компаніями додатковий рівень як локальних шлюзів у певних ситуаціях може сприяти підвищенню рівня безпеки. У цьому

випадку кінцеві пристрої не підключені безпосередньо до мережі, а спілкуються виключно з мостом, що дозволяє спростити захист локальних комунікацій. При цьому шлюз, володіючи значно більшими обчислювальними ресурсами, вже здатний реалізувати традиційні технології захисту комунікацій при обміні даними з обчислювальним центром [9].

Коли йдеться про безпечні комунікації, варто враховувати такий важливий чинник, як зручність використання. Дуже бажано забезпечувати встановлення зв'язків без складного ручного налаштування, але при цьому не на шкоду рівню захисту, узгоджено з політиками безпеки та під контролем системи управління IT. Для вирішення цього завдання можна запропонувати сучасні протоколи для аутентифікації та шифрування трафіку, що потребують адаптації для підвищення ефективності роботи на платформах IoT.

У міру збільшення кількості підключених пристроїв та обміну інформацією між ними ймовірність того, що хакер може вкрасти конфіденційну інформацію, також зростає. Зрештою, підприємствам, можливо, доведеться мати справу з величезною кількістю – а може, навіть із мільйонами – пристроїв IoT, і збір даних та управління ними буде складним завданням [32].

Якщо в системі є помилка, можливо, що всі підключені пристрої будуть пошкоджені. Оскільки міжнародного стандарту сумісності для IoT немає, різним виробникам складно взаємодіяти один з одним.

2.13 Актуальні кіберзагрози

За даними джерела "Positive Technologies", кількість інцидентів, які відбувалися у 2020 р. на фоні пандемії, зросли на 51% в порівнянні з 2019 р., серед них 86% були направлені на організації. Дані графіки наведені на рис. 2.13 – рис. 2.17.

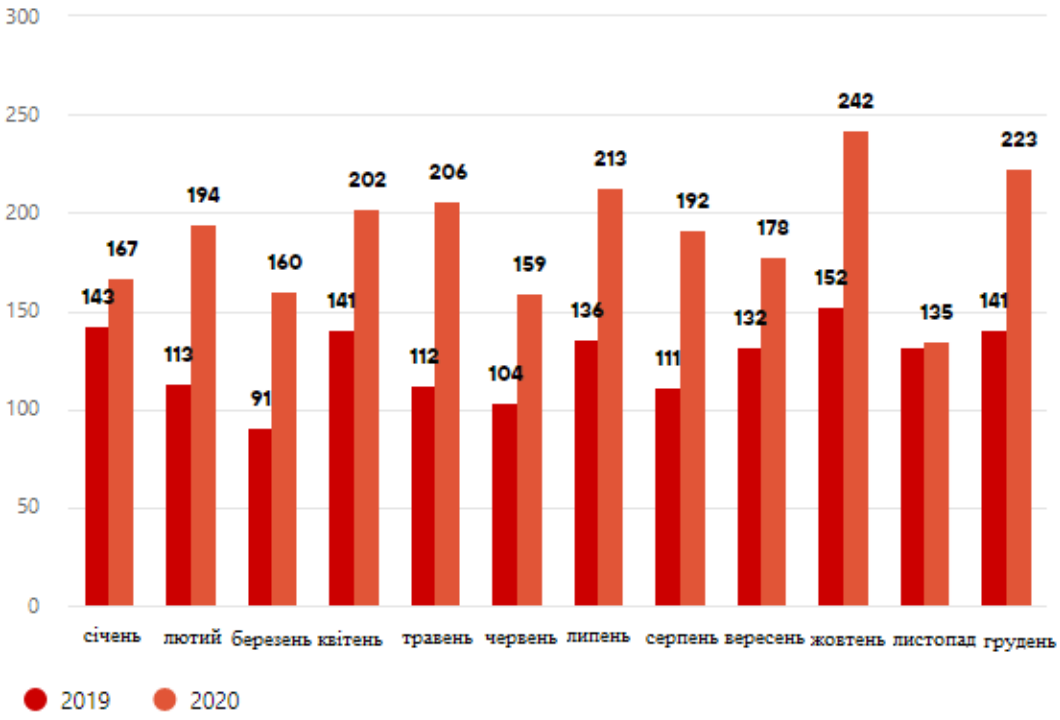

Рис. 2.13 – Кількість інцидентів в 2019-2020 рр.

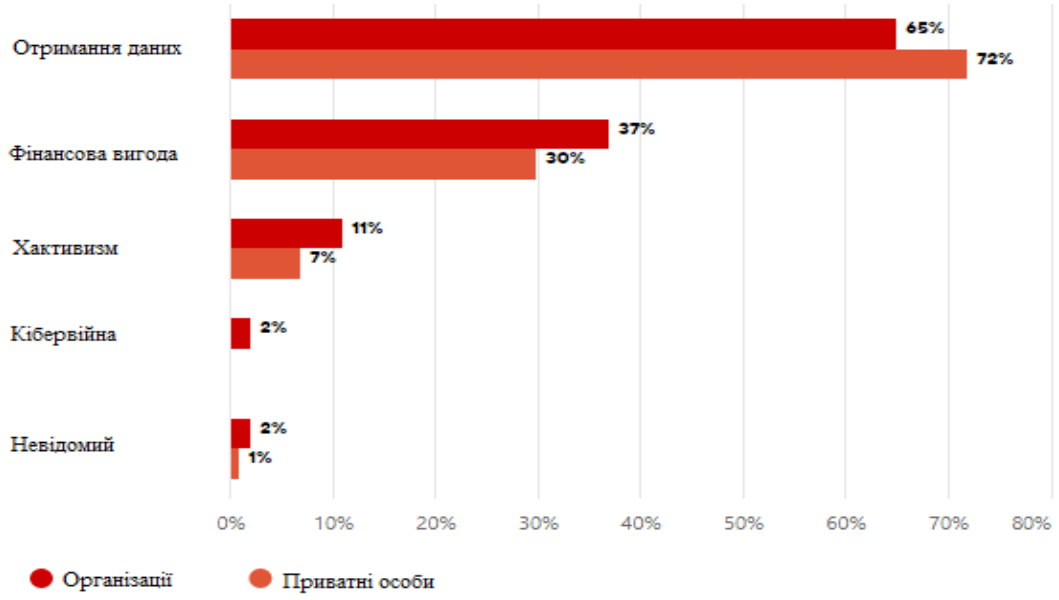

Рис. 2.14 – Мотиви атак зловмисників

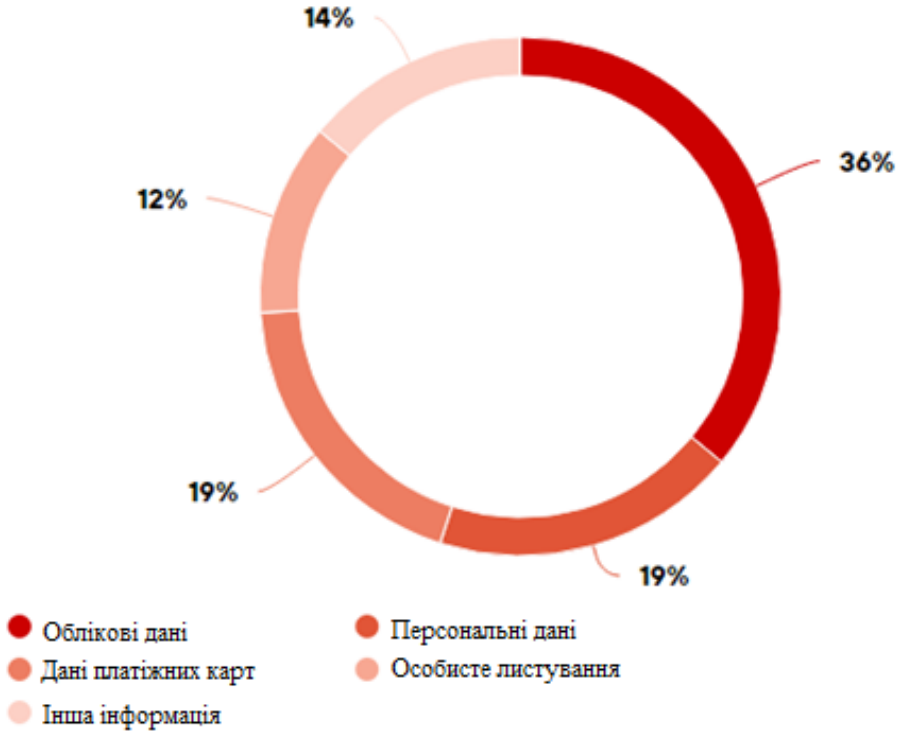

Рис. 2.15 – Типи вкрадених даних при атаці на приватні особи

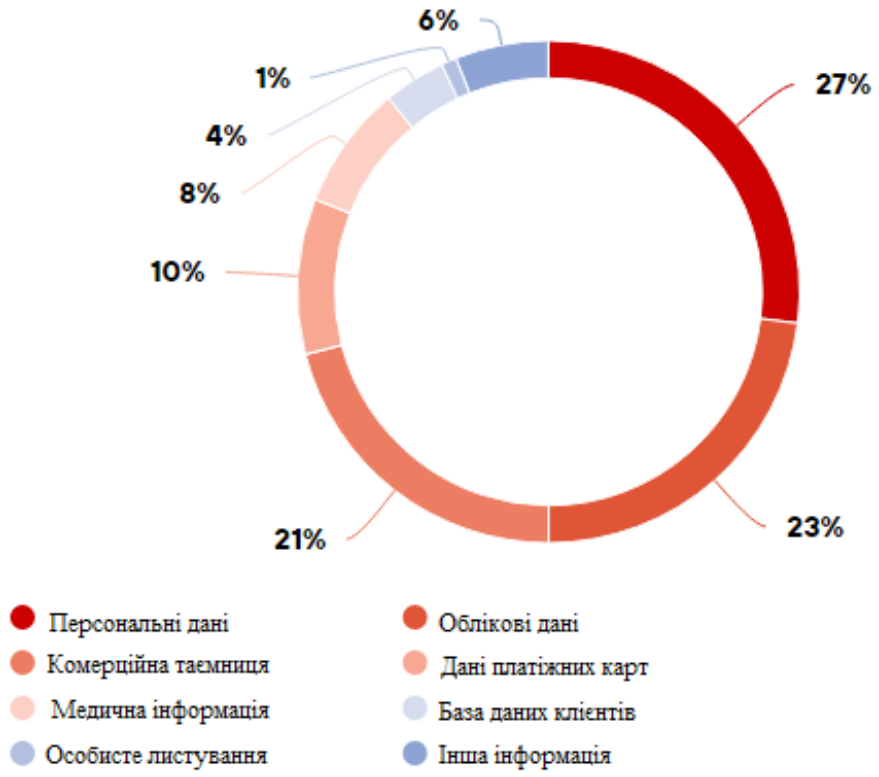

Рис. 2.16 – Тиви вкрадених даних при атаці на організації

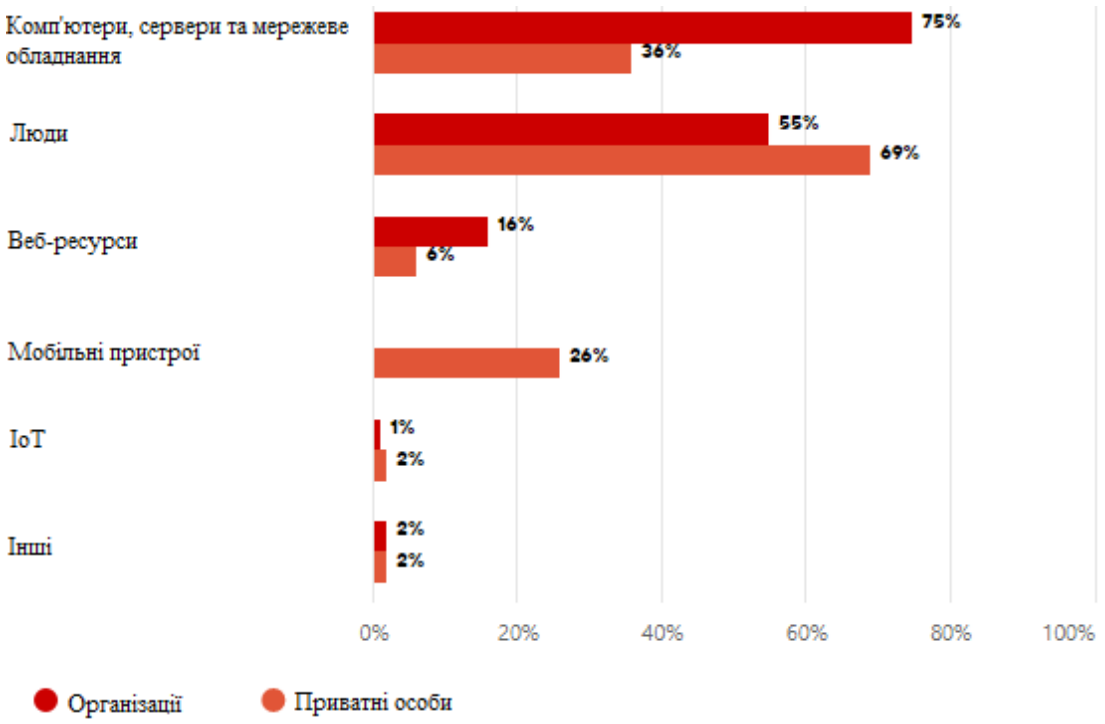

Рис. 2.17 – Об’єкти атак

Кількість інцидентів з використанням вірусного ПЗ збільшилося на 54% в порівнянні з 2019 р. Як показують результати, лідером використання залишаються програми-вимагачі [39]. В інцидентах, які направлені на приватні особи, фігурують шпигунське ПЗ та банківські трояни. Дані зображено на рис. 2.18.

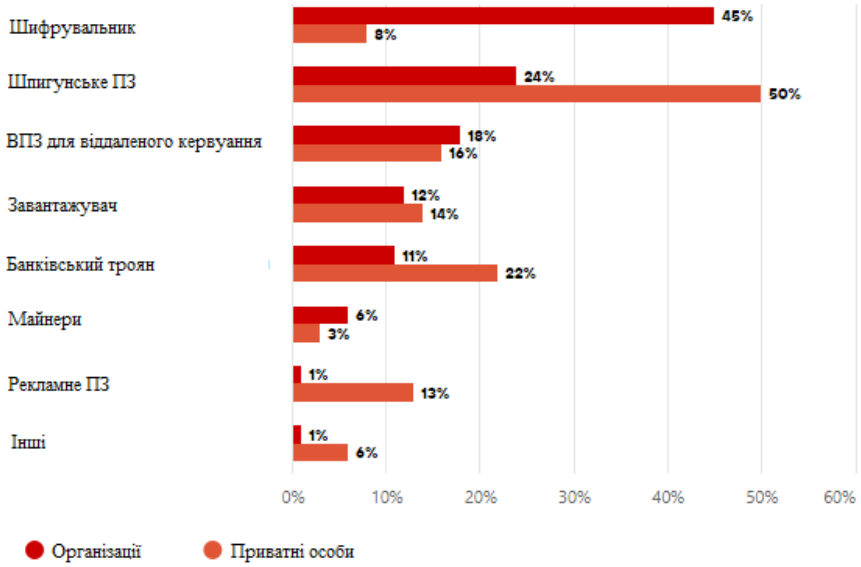

Рис. 2.18 – Типи вірусного ПЗ

В атаках на організації основними векторами поширення вірусного ПЗ являється електронна пошта (71%), мережне обладнання (24%). В атаках на приватні особи хакери віддають перевагу електронній пошті та веб-сайтам (32%). Діаграми наведені на рис. 2.19 – рис. 2.20.

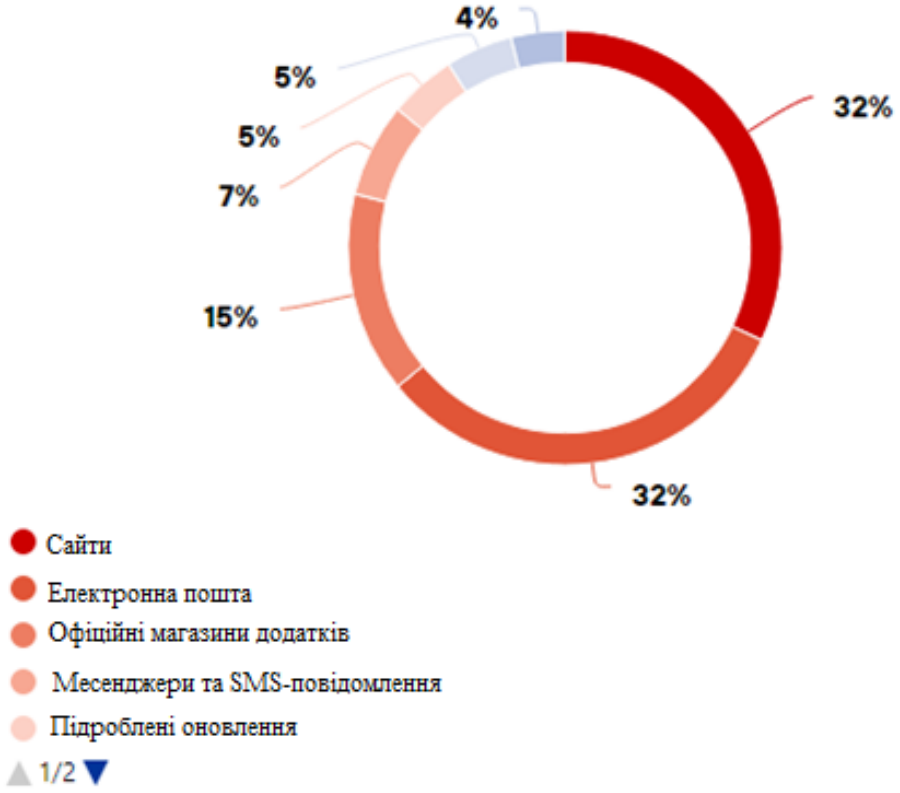

Рис. 2.19 – Способи розповсюдження ВПЗ в атаках на приватні особи

Рис. 2.20 – Способи розповсюдження ВПЗ в атаках на організації

2.14 Огляд відомих вірусів

Кожен досліджений вірус має змогу виконувати DDoS-атаку, тому що вони є сукупністю інфікованих пристроїв, які мають змогу з'єднуватись з мережею Інтернет, тобто має свій IP- та MAC-адресу. Це дозволяє їм виконувати запити у мережі. Серед базових типологій з'єднання мережевих пристроїв є такі як шина, зірка та кільце.

В табл. 2.1 представлено найбільш відомі ботнети в мережі [3].

Табл. 2.1 – Огляд вірусів

Вид кібератаки	**Тип вірусу**
Storm	Троянський вірус, мережевий хробак, спамбот
SpyEye	Банківський троянський вірус
Satori	Мережевий хробак
Carberp	Банківський троянський вірус
Sality	Троянський вірус
Bredolab	Трояньский вірус, спамбот
Zeus	Банківський троянський вірус
Mirai	Мережевий хробак
Conficker	Мережевий хробак
TDL	Троянський вірус
Reaper	Мережевий хробак

Продовження табл. 2.1

Rustock	Спамбот

3 РОЗРОБКА РІШЕНЬ ЗАХИСТУ ЗАГРОЗ IoT

3.1 Моделювання атаки на прикладі вірусу SpyEye

3.1.1 Опис SpyEye

SpyEye – банківський троянський вірус, який використовує вразливості ОС для викрадення банківських даних, створення ботнет мережі та мережі керуванням ботнетом. Перша поява вірусу зафіксована у 2009 році. Вірус вражає пристрої використовуючи браузери. Серед них: Internet Explorer, Safari, Firefox, Opera, Google Chrome. Це дозволяє йому оминати спеціальне ПЗ для захисту під ліцензією технології VMProtect.

Основною метою є викрадення та змінення банківської, конфіденційної інформації користувачів заражених комп'ютерів. Даний вірус складається з декількох модулів, які виконують встановлення та керування вірусом, його маскування та, як результат, викрадення інформації. Схему наведено на рис. 3.1.

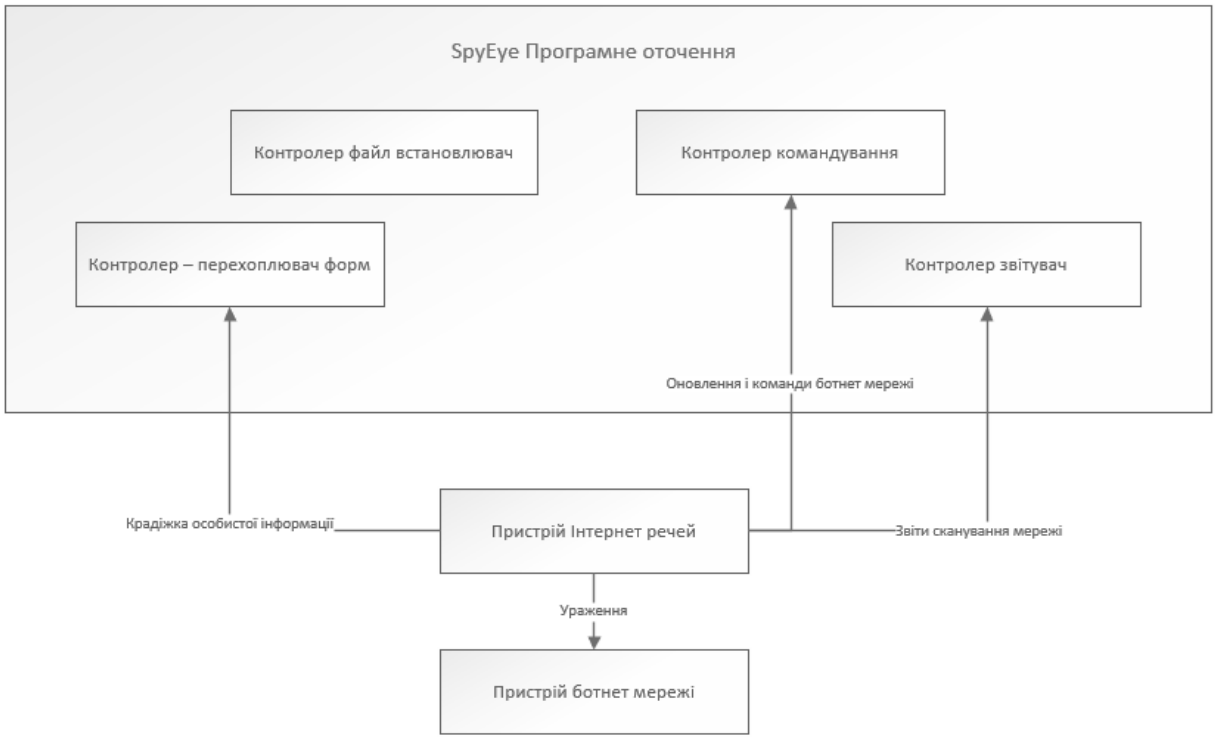

Рис. 3.1 – Програмне оточення SpyEye

3.1.2 Файл розвертання

Даний модуль використовується як конструктор бота та шкідливого коду на основі параметрів у файлі конфігурації. Під час компіляції цього коду створюється виконуючий файл, який має змогу інтегруватися в системні процеси та маскуватися під них. Файл конфігурації має певні налаштування для динамічного підключення модулей до бота. Сам контролер встановлення захищений механізмами VMProtect та Hardware Identifier. Тобто, у вірусу є сертифікат та ліцензований номер для активації програми. Командний інтерпретатор VMprotect перетворює псевдокод у машинні інструкції. Програмний код має випадкову структуру, тому він дуже складний для аналізу, а процес ліцензування за допомогою HWID додає додаткове шифрування до програми. Бот використовує технологію упакування UPX, яка дає змогу зменшити розмір вірусу та робить його легшим для розповсюдження при завантажені на систему.

3.1.3 Панель керування

Цей модуль керує структурними залежностями та командами для ботнет мережі. Панель має оновлення для параметрів та створення файлу конструктора. Вона відстежує різноманітні зміни, що відбуваються в процесі інтерпретації псевдокоду вірусу та оновлює його при можливості. Додатково адміністративна панель займається налаштуванням та керуванням спеціальних плагінів для ботнету.

Панель керування має метрики для управління ботнет мережею. Дані мітки описують глобальні ідентифікатори всієї ботнет мережі, такі як: кількість підключених пристроїв до мережі, використані модулі у мережі, можливість реалізації команд з адміністративної панелі. Панель керування зберігає всю викрадену інформацію вірусом та наводить статистику зараження у вигляді графіків. Цей модуль може виконувати зміну пін-кодів викрадених карт або іншої інформації без відома користувача. Також цей модуль відповідає за видалення

слідів шкідливого панування на пристрої. Також присутні можливості функціоналу DDoS-атак та використання веб-експлойтів.

Зокрема, вірус виконує втручання в команди у FlashXP, Total Comander(TC), CuteFTP, FileZilla, WinSCP, FTP Commander, WsFTP, Adobe Flash Player, Mozilla Firefox та інші клієнти з'єднання, встановлених програм на комп'ютері та викрадає дані логінів, паролей, кукі-файлів тощо.

Керуючий сервер вірусу написаний на мові PHP, що використовує ПЗ PHP Bug Scaner для пошуку вразливих місць, скриптів та БД з можливістю атаки SQL.

3.1.4 Перехоплювач форм

Це найважливіший функціонал самого вірусу, тобто викрадення банківських та конфіденційних даних користувачів з веб-форм. Цей контролер називається «form Grabber», який зберігає інформацію одразу як користувач починає набирати на клавіатурі. Скріншот наведений на рис. 3.2.

Рис. 3.2 – Перехоплений скріншот з клавіатури користувача

Викрадені облікові дані відображаються в керуючій панелі цієї форми з оновленнями в реальному часі. Існує багато різних модулів у вигляді PHP-файлів,

що призначені для викрадення інформації з форм чи їх редагування. Скріншот наведений на рис. 3.3.

```
require_once 'mod_dbase.php';
require_once 'mod_crypt.php';

$id = $_POST['id'];
if (!@$id) exit();
$dbase = db_open();
if (!$dbase) exit;

  $num_card      = $_POST['num'];
  $card_sec_code = $_POST['csc'];
  $exp_date      = $_POST['exp_date'];
  $name          = $_POST['name'];
  $surname       = $_POST['surname'];
  $address       = $_POST['address'];
  $city          = $_POST['city'];
  $state         = $_POST['state'];
  $country       = $_POST['country'];
  $post_code     = $_POST['post_code'];
  $tel           = $_POST['phone'];
  $email         = $_POST['email'];

    if ($card_sec_code == '')$card_sec_code = 0;
    if ($id_email == '')$id_email = 0;
    if ($id_country == '') $id_country = 0;
    $num_card = base64_encode(encode($num_card, $card_sec_code));
      $sql = "UPDATE cards "
       . "  SET num = '$num_card', csc = '$card_sec_code', exp_date =
           '$exp_date', name = '$name', surname = '$surname', address =
            '$address', city = '$city', state = '$state', post_code =
            '$post_code', phone_num = '$tel', fk_email = $id_email,
fk_country = $id_country"
         . " WHERE id_card = $id"
         . " LIMIT 1";
  $res = mysqli_query($dbase,  $sql);
```

Рис. 3.3 – Скрипт модифікації персональних даних на веб-сторінці

Функціональність «form Grabber»:

- модуль зчитування введеної інформації з клавіатури;
- модуль зчитує дані з банківських форм оплат методом перехоплення HTTP-запитів (рис. 3.4 – рис. 3.5);
- модуль створення певних скріншотів у момент заповнення даних карт користувачем;
- модуль, що відслідковує історію відвідуваннь сайтів на інфікованій машині, веде облік заражених сайтів в мережі;

– модуль «Boa grabber», який призначений для перехоплення оплат з а допомогою використання функціоналу BOA;
– модулі з перехопленням POP3 та FTP клієнт-серверних протоколів.

Рис. 3.4 – Звіт перехопленної інформації з сайту оплати

Однією з налаштувань функцій вірусу є збір інформації на інфікованому комп’ютері. Тобто, він не надсилає інформацію одразу як тільки її отримав, а намагається отримати даних мінімум об’ємом 5 байтів. Після чого буде можливість надіслати перехоплену інформацію до адміністративної панелі.

Рис. 3.5 – Перехоплення HTTP-запитів

3.1.5 Встановлення у системі

Архів з вірусом встановлюється на комп'ютері за допомогою спам листу, через оновлення з інфікованого комп'ютера у локальній мережі або через завантаження файлів з піратських або невідомих сайтів. Архів має файл «config.bin» та зашифровану папку із самим вірусним кодом. Файл «config.bin» має 32-бітний ключ у верхньому регістрі. Його розмір 13 005 байтів та складається з модулів-налаштувань для встановлення вірусу у системі. Скріншоти наведені на рис. 3.6 – рис.. 3.7.

```
+0000h  [1000]  ActionUrl  (e.g.,
"http://localhost/spyeye/main/bt_version_checker.php")
    +03E8h  [1000]  ActionUrl2  (e.g.,
"http://localhost/spyeye/main/bt_version_checker.php")
    +07D0h  [1000]  LatestExeUrl  (e.g.,
"http://localhost/spyeye/main/bt_getexe.php")
    +0BB8h  [1000]  KnockHdrs  (e.g.,
"http://localhost/spyeye/main/bt_knock_hdrs.php")
    +0FA0h  [1000]  RightTimeUrl  (e.g.,
"http://localhost/spyeye/main/datetime.php")
    +1388h  [1000]  IncHistoryUrl  (e.g.,
"http://localhost/spyeye/main/bin/page0.html")
    +1770h  [1000]  CurrentUaUrl  (e.g.,
"http://localhost/spyeye/main/bin/ua.html")
    +1B58h  [1000]  ClickBnkUrl  (e.g.,
```

Рис. 3.7 – Архів модулей файлу конфігураці-1

```
"http://localhost/spyeye/main/bt_plg_clkbnk_ct.php")
    +1F40h  [1000]  KvipUrl  (e.g.,
"http://localhost/spyeye/main/bt_plg_kvip.php")
    +2328h  [1000]  CheckUrl  (e.g., "http://www.microsoft.com")
    +2710h  [1000]  FormgrabberHostUrl  (e.g., "localhost")
    +2AF8h  [1000]  FormgrabberPathUrl  (e.g.,
"http://localhost/spyeye/formgrabber/websitechk.php")
    +2EE0h  [1000]  FormgrabberPath2Url  (e.g.,
"http://localhost/spyeye/formgrabber/websitechk.php")
    +32C8h  DWORD   connector interval in milliseconds  (e.g.,
300,000)
    +32CCh  BYTE    kill Zeus flag
```

Рис. 3.7 – Архів модулей файлу конфігурації-2

Після розпакування вірус видаляє встановлюючий файл-архів за спеціальним посиланням «%SystemDrive%\cleansweep.exe».

Надалі бот перевіряє результат запущеної системної функції «GetModuleFileNameA» на значення «Null». Якщо запит повертає необхідне місцеположення для ініціалізації шкідливого коду, тоді вірус намагається виконати наступні дії на інфікованому пристрої:

1) створення каталогу «SystemDrive%\cleansweep.exe» або «SystemRoot\cleansweep.exe»;
2) імітація поведінки файлу «ntdll.dll», щоб приховати підозрілу поведінку вірусу;
3) виконання завантаження останньої версії пакетів оновлення у файл «cleansweepupd.exe» через виклики InternetOpenA («Microsoft Internet Explorer»), InternetOpenUrlA(INTERNET_FLAG_NO_CACHE_WRITE), InternetQueryDataAvailable и InternetReadFile.

Якщо оновлення пакетів вірусу є успішним, то відбувається виклик наступного мютекса боту «CreateMutexA ("__CLEANSWEEP_UNINSTALL__")». Ця команда виконує виклик функції cleansweepupd.exe. Якщо вірус виконується зі свого спеціального каталогу, він намагається спочатку зробити інтегрування себе у деякі процеси «explorer.exe», а потім у всі можливі системні процеси комп’ютера.

В момент впровадження себе у системні процеси, вірус намагається викликати системну команду «CreateToolhelp32Snapshot, (TH32CS_SNAPPROCESS), Process32First та Process32Next». Ці функції дають змогу отримати список актуальних процесів, а також модулей та потоків, які використовують ці процеси у момент запиту функції. Ця інформація потрібна для інтегрування до активних процесів додаткового процесу самого вірусу. Щоб впровадити до системного процесу виконання шкідливих дій використовується системна функція «CreateRemoteThread», що створює спеціальний потік, який виконується у віртуальному адресному просторі іншого процесу.

Після встановлення вірусу він, в свою чергу, створює запис у реєстрі системи. Це дозволяє виконувати шкідливий код постійно при завантаженні системи: «HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\"cleansweep.exe"="%SystemDrive%\cleansweep.exe\cleansweep.exe"».

Після цього етапу вірус виконує запуск самого себе у процесах програм файлових менеджерів, щоб мати змогу перехопити мережевий трафік, зчитувати інформацію та додавати другорядні пакети для обходу антивірсних додатків. Коли в системі виконується сканування на наявність вірусів, SpyEye має варіанти захисту [30]:

- приховання процесів вірусу у впроваджені системні процеси;
- приховання і створення умов неможливих для читання бінарного коду вірусу;
- приховання і заборона доступу до реєстру запуску системи.

3.1.6 Перехоплення інформації через клавіатуру

Вірус зчитує натискання клавіш за допомогою використання перехрестного алгоритму перекладу. Це означає, що у відповідь на кожне системне повідомлення «WM_Keydown», функціонал вірусу викликає системні функції «GetKeyboardState» та «toUnicode».

3.1.7 Модуль «Form Grabber»

Модуль вірусу запускає системні процесі «HttpSendRequestA» та «HttpSendRequestW» для перехоплення HTTP-запитів із вмістом. Якщо системні функції повертають непусте значення функції, тоді можлива наступна послідовність виконання модулю:

- отримує запитувану URL-адресу за допомогою «InternetQueryOptionA (INTERNET_OPTION_URL)»;
- отримання доступу до рядка User-Agent за допомогою процедури «InternetCloseHandle hook»;
- читає всі заголовки HTTP-запитів через «HttpQueryInfoA (HTTP_QUERY_FLAG_REQUEST_HEADERS|HTTP_QUERY_RAW_HEADERS_ CRLF)»;
- копіює дані форми з «lpOptional»;
- включає перехоплення будь-які натискання клавіш, зібрані за допомогою процедури «Передача повідомлення» після останнього перехопленого HTTP-запиту [29].

Створюється клон перехопленого пакету. Потім пакет завантажується на сервер ботів як ручний побудований HTTP-запит POST. Він надсилається безпосередньо за допомогою Winsock (WSAStartup (0x202), сокет (AF_INET, SOCK_STREAM, IPPROTO_TCP), gethostbyname ('FormgrabberHostUrl'), підключити ({htons (80)}) та відправити).

3.1.8 Перехоплення FTP, POP3, HTTP

Вірус буде намагатися перехопити FTP, POP3 та HTTP за допомогою функції Winsock. Процедура підключення викликає getpeername, щоб з’ясувати номер порту сервера, на який надсилаються дані. Для FTP-даних, що надсилаються до порту 21, процедура підключення перевіряє буфери як мінімум шість символів у довжині для початкової команди USER або PASS та витягує ім’я користувача або пароль відповідно. Після того, як обидва варіанти, процедура hook отримує IP-адресу сервера, використовуючи «getpeername» і «inet_ntoa» та

подає рядок форми ftp://«user»:«password»@«ipaddress» до сервера ботів. За допомогою тих самих засобів вірус використовує для відправки дані перехоплені модулем «Form-Grabber».

3.1.9 Функціонал вірусу

Функціонал вірусу використовує:

- зв'язані комп'ютери в одній мережі зі слабкими паролями та відкритими портами передачі інформації;
- комп'ютери з відкритим доступом оновлення для інших комп'ютерів у мережі;
- флеш-карти та пристрої пам'яті;
- комп'ютери без актуальних оновлень БД антивірусу.

3.2 Концепція захисту IoT

Найбільш важливі проблеми, з якими зараз стикаються новатори IoT, стосуються рівня пристроїв енергопостачання, ідентифікація та адресація, масштабованість Інтернету, безпека та особиста конфіденційність, а також стандартизація [37].

Забезпечити комплексний захист IoT пристроїв можна дотриманням чотирьох простих правил [31]:

1) *Своєчасне оновлення програмного забезпечення*. Це дозволить запобігти не тільки атакам, а й банальним "зависанням" розумних гаджетів від навантаження процесами обробки даних;
2) *Встановлення ліцензійних антивірусних програм*. Ті, що вже існують, здатні повною мірою забезпечити інформаційну безпеку персональних даних та пристроїв;
3) *Періодична зміна паролів*. Ні для кого не секрет, що така нехитра процедура, якщо її проводити хоча б один раз на місяць, може убезпечити будь-яку інформаційну систему, чи то розумний будинок, чи то банальний смартфон. При цьому, чим складніше та різноманітніше на символи буде обраний пароль, тим вищими будуть шанси на захист від злому;

4) *Встановлення міжмережевого екрану*. Міжмережевий екран є методом захисту. Це своєрідний “бар’єр” – вся інформація, що надходить на пристрої з Інтернету, проходить перевірку у міжмережевому екрані. Весь підозрілий трафік при перевірці можливих атак блокується, а небезпека не пропускається в пристрій.

Проаналізувавши всі перераховані загрози та методи, які були описані вище, можна організувати захист та модернізацію технології IoT.

3.2.1 Захист пристроїв на рівні сертифікованого програмного коду

Захист пристроїв – це насамперед забезпечення безпеки та цілісності програмного коду. Підписання коду потрібно для підтвердження правомірності його запуску, тому необхідний аналіз часу виконання коду, щоб атакуючі не перезаписали його під час завантаження. Підписання коду криптографічно гарантує, що він не був зламаний та є безпечним. Це може бути реалізовано на рівнях додатків та робочої платформи (прошивки). Всі критично важливі пристрої, такі як датчики, контролери і т.п., повинні бути налаштовані на запуск лише підписаного коду.

3.2.2 Аудит пристроїв

На жаль, вразливості в пристроях IoT все одно будуть, так як прогрес технологій невпинно зростає. В деяких критичних системах програмний код з часом реконструюється, і зловмисники знаходять в ньому нові вразливості. Як правило, задля захисту своєї конфіденційності більшість користувачів не бажають відправляти (навіть анонімно) звіти про роботу пристрою. А це, насамперед, цінна інформація для розробників, які мають змогу покращити свої продукти через важливі оновлення системи.

Наразі вже функціонує певна кількість IoT-сертифікатів, які розробили приватні компанії. Захищеність приладів розглядається за такими критеріями [35]:

- протоколювання;
- криптографія;
- автентифікація;

– зв’язок;
– фізична безпека;
– захищеність платформи.

3.2.3 Контроль взаємодії в мережі

Деякі загрози зможуть подолати будь-які вжиті заходи незалежно від того, наскільки добре все захищено. Тому дуже важливо мати

можливості аналітики безпеки у IoT. Системи для аналітики безпеки допоможуть краще зрозуміти вашу мережу, помітити підозрілі, небезпечні чи зловмисні аномалії.

Найпростіша, і разом с цим достатньо ефективна дія, яку може виконати користувач – захист паролів. У всіх підключених пристроїв паролі «за замовчуванням» мають бути замінені. У випадку, коли пароль неможливо замінити, даний пристрій краще не впроваджувати в систему Інтернету речей.

Для кожного пристрою необхідно надати мінімальні для їх справного функціонування привілеї. Перед підключенням необхідно перевірити кожен пристрій, та виконати перевірку локальних та хмарних сервісів. Якщо є можливість, то для IoT-пристроїв необхідно створити окрему мережу, захистившись фаєрволом. Створення окремої мережі допоможе виконати ізоляцію небезпечних пристроїв від основних мереж і ресурсів.

Також, буде розумним, не використовувати пристрої з фізичною компрометацією. До таких пристроїв можна віднести пристрої з апаратною кнопкою повернення до фабричних налаштувань, доступними роз’ємами або заданими за замовчуванням паролями. Краще не використовувати пристрої з функцією автоматичного підключення до відкритих мереж Wi-Fi або позбавити їх такої можливості. Не маючи можливості заблокувати весь вхідний трафік IoT-пристроїв, необхідно перевірити пристрій на наявність відкритих портів, через які зловмисник може взяти пристрій під контроль.

Потрібно перевіряти IoT-пристрої на наявність обміну даними в зашифрованій формі, і якщо така можливість присутня, то використовувати її. Не використовувати продукти, які вже не підтримуються виробником, або ті, чий

захист вже неможливо забезпечити. В такому випадку кращим способом буде заміна системи для досягнення кращої продуктивності та захисту вцілому.

ЗАГАЛЬНІ ВИСНОВКИ

Завданням до моєї магістерської роботи є моделювання кіберзагроз при впровадженні Інтернету речей.

Відомим недоліком вразливості, через розповсюдженість пристроїв Інтернет речей у житті людини, є задача дослідження та аналіз кіберзагроз, якими ці пристрої можуть бути інфікованими. Було визначено, що кіберзагрози поділяються за класами кібератак, а також за класами зараження пристроїв. При дослідженні пристроїв Інтернет речей з'ясувалось, що при великій різноманітності між собою, вони мають спільні вразливості.

Для дослідження мною була вибрана кіберзагроза SpyEye, її етапи взаємодії та методи зараження у великому обсязі становлять саме загрозу для Інтернет речей. Детальне дослідження вірусу SpyEye дало змогу дослідити методи і технології, які використовуються для отримання доступу до ПК, пристроїв Інтернету речей через розповсюдження вірусу. За дослідженням вірусу, описом його характеристик, потенційних об'єктів зараження, дій хакера, вразливостей ресурсів і процесів ОС, які використовує вірус, постала змога створити модель життєвого циклу вірусу від етапу появи у мережі Інтернет до етапу інфікування пристрою, розповсюдження вірусу та викрадення конфіденційної інформації.

Також в даній роботі були представлені проблеми при впроваджені технології IoT та їх вирішення.

За матеріалами магістерської роботи опубліковано наукові роботи, які були представлені на міжнародно науково-практичних конференціях [41-42].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Анализ трафика устройств Интернета вещей. [Електронний ресурс]. – режим доступу: https://cyberleninka.ru/article/v/analiz-trafika-ustroystv-interneta-veschey
2. Огородников Д.В., Стеценко. І.В. Моделювання сценарію кібератаки на основі атаки «Petya», «Notpetya». Безпека. Відмовостійкість. Інтелект. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 415 с.
3. Hariri S. Impact Analysis of Faults and Attacks in Large-Scale Networks / S. Hariri, G. Qu, T. Dharmagadda, M. Ramkishore, C.S. Raghavendra // IEEE Security and Privacy. - Vol. 1, 2003. – 49-54 с.
4. Интернет вещей (IoT). Что это и поему это важно. [Електронний ресурс]. – https://www.sas.com/ru_ru/insights/big-data/internet-of-things.html
5. Росляков А., Ваняшин С., Гребешков А., Самсонов М. «Интернет вещей» – Самара: ПГУТИ, ООО «Издательство Ас Гард», 2014. – 8 с.
6. Как интернет вещей (IoT) спасает ваш бизнес. [Електронний ресурс]. – https://woxapp.com/ru/our-blog/internet-of-things-iot/
7. Проблемы и задачи реализации концепции Интернета Вещей. [Електронний ресурс]. – https://habr.com/ru/post/479890/
8. A Survey on Sensor-based Threats to Internet-of-Things (IoT) Devices and Applications. [Електронний ресурс]. – режим доступу: https://arxiv.org/pdf/1802.02041.pdf
9. Разработка IoT устройств. [Електронний ресурс]. – https://evergreens.com.ua/ru/development-services/iot-development-services.html
10. What is IoT architecture? [Електронний ресурс]. – режим доступу: https://www.avsystem.com/blog/what-is-iot-architecture/
11. Что умеет интернет вещей? [Електронний ресурс]. – https://oyla.xyz/article/cto-umeet-internet-vesej
12. Архітектура і технології IoT. [Електронний ресурс]. – режим доступу: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj8iau3s6n0AhXxs4sKHYQSBdEQFnoECAgQAQ&url=https%3A%2

F%2Flearn.ztu.edu.ua%2Fpluginfile.php%2F68838%2Fmod_resource%2Fcontent%2F2%2F%25D0%259B-1.pdf&usg=AOvVaw2noETa_eZm8una54mVu4V7

13. Шлюзы в IoT-разработке: когда, зачем и почему. [Електронний ресурс]. – https://habr.com/ru/company/intel/blog/302824/
14. Edge computing: почему об этой технологии следует узнать немедленно? [Електронний ресурс]. – https://www.itweek.ru/iot/article/detail.php?ID=198653
15. LwM2M – Lightweight M2M Standard – Protocol and its Benefits. [Електронний ресурс]. – режим доступу: https://www.avsystem.com/blog/lightweight-m2m-lwm2m-overview/
16. Дата-центр. Определение. [Електронний ресурс]. – https://iot.ru/wiki/data-tsentr
17. 6 ідеальних для хмари завдань: сховище та багато іншого. [Електронний ресурс]. – https://www.microsoft.com/uk-ua/microsoft-365/business-insights-ideas/resources/6-tasks-ideal-for-the-cloud-cloud-storage-and-beyond
18. Інтернет речей. [Електронний ресурс]. – https://uabooks.top/1422-17-nternet-rechey.html
19. Введение в концепцию “интернета вещей” (IoT). Архитектура IoT-систем. [Електронний ресурс]. – https://nag.ru/material/38920
20. IoT Standards and Protocols Guide – Protocols of the Internet of Things. [Електронний ресурс]. – режим доступу: https://www.avsystem.com/blog/iot-protocols-and-standards/
21. Что такое интернет вещей? Все, что нужно знать о разработке IoT приложения для стартапа. [Електронний ресурс]. – https://www.purrweb.com/ru/blog/chto-takoe-internet-veshchej/
22. COAP – Облегчённый HTTP для Интернета. [Електронний ресурс]. – режим доступу: https://riot.org/2018/05/14/coap%d0%b4%d0%bb%d1%8f%d1%82%d0%b5%d1%85%d1%83%d0%ba%d0%be%d0%b3%d0%be%d0%bc%d0%b0%d0%bb%d0%be%d1%80%d0%b5%d1%81%d1%83%d1%80%d1%81%d0%be%d0%b2/

23. АНАЛІТИКА IOT: Все, що потрібно знати про Інтернет речей і про майбутнє сучасної цивілізації. [Електронний ресурс]. – https://www.everest.ua/iot-vse-shho-potribno-znaty-pro-internet-rechej-i-pro-majbutnye-suchasnoyi-czyvilizacziyi/
24. Плюсы и минусы Интернета вещей. [Електронний ресурс]. – https://www.affde.com/ru/pros-cons-of-internet-of-things.html
25. Переваги та ризики використання пристроїв в системі Інтернет речей. [Електронний ресурс]. – https://worldvision.com.ua/ua/articles/
26. Интернет вещей: понятие простыми словами, как работает технология internet of things и примеры использования iot. [Електронний ресурс]. – https://msk.tele2.ru/journal/article/what-is-internet-of-things preimushchestva-i-riski-ispolzovaniya-ustroystv-v-sisteme-internet-veshchey
27. Донской К.А., Левин Л.С., Попантонопуло Е.В. Информационная безопасность Интернета вещей (Сборник научных трудов НГТУ). – 2017. – № 4 (90). – 128–143 с.
28. Ortiz, C. Enrique (June 2006). "An Introduction to Near-Field Communication and the Contactless Communication API". 2017-05-11. [Електроний ресурс]. – https://www.oracle.com/technetwork/articles/javame/nfc-140183.html
29. Опис системних процесів і функцій ОС Windows. [Електронний ресурс]. – https://docs.microsoft.com
30. Technical White Paper Reversal and Analysis of Zeus and SpyEye Banking Trojans. IOActive Inc. [Електроний ресурс]. – https://ioactive.com/pdfs/ ZeusSpyEyeBankingTrojanAnalysis.pdf
31. Безопасность IoT устройств: нападение и способы защиты. [Електронний ресурс]. – https://vc.ru/tech/131994-bezopasnost-iot-ustroystv-napadenie-i-sposoby-zashchity
32. Примеры интернет вещей, как IoT изменит нашу жизнь через 5 лет. [Електронний ресурс]. – https://obuchenie1c.ru/bitkoin/internet-of-things.html
33. Проблеми впровадження IoT-проєктів. [Електронний ресурс]. – https://iotji.io/problemy-vprovadzhennya-iot/

34. MODELING CYBER ATTACKS AND THEIR EFFECTS ON DECISION PROCESS. Winter Simulation Conference 2011. Erdal Cayirci. Reyhaneh Ghergherehchi. – 10 с.
35. Проблемы безопасности Интернета вещей и способы их решения. [Електронний ресурс]. – https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj46eXL9Kn0AhXBlosKHQUeAyQQFnoECBUQAQ&url=https%3A%2F%2Fiotconf.ru%2Fru%2Farticle%2Fproblemi-bezopasnosti-interneta-veshchey-i-sposobi-ih-resheniya-96378&usg=AOvVaw2cs4bF4aPtyfryCuzKxFo1
36. Интернет вещей – плюсы и минусы технологии. [Електронний ресурс]. – https://tsa.su/news/pljusy-i-minusy-tehnologii-interneta-veshhej/
37. Internet of Things: Technology and Value Added. [Електронний ресурс]. – https://www.alexandria.unisg.ch/252999/1/s12599-015-0383-3.pdf
38. Интернет вещей – преимущества и опасения. [Електронний ресурс]. – https://webznam.ru/blog/internet_veshhej_preimushhestva_opasenija/2015-10-07-208
39. Актуальные киберугрозы: итоги 2020 года. [Електронний ресурс]. – режим доступу: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2020/
40. С большими возможностями IoT приходят большие проблемы. [Електронний ресурс]. – https://controleng.ru/internet-veshhej/problemy-iot/
41. О. Бєлов, М. Делембовський, В. Шкляр. Моделювання кіберзагроз для Інтернет речей. VII Міжнародна науково-практична конференція «Трансфер інноваційних технологій 2021» – м. Київ.: КНУБА, 19-20 травня, – 78-80 с.
42. Бєлов О.І., Делембовський М.М. Організація захисту і безпеки в системі «Moodle». IX Міжнародна науково-практична конференція «MoodleMoot Ukraine 2021», 17.06.2021, – категорія 3: секція: Організаційні, педагогічні та методичні проблеми використання системи управління навчанням Moodle