Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Аналіз механізмів безпеки операційної системи Home Assistant"

Виконав: студент

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)

Слободян П.П.
підпис (прізвище та ініціали)

Керівник Деркач М.В.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д.І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Слободян Поліні Петрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз механізмів безпеки операційної системи Home Assistant

Керівник роботи Деркач Марина Володимирівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «15» 04 2024 року № 4/7-350

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Технічна документація Home Assistant

4. Зміст роботи (перелік питань, які потрібно розробити)
Галузь застосування операційної системи Home Assistant
Механізми безпеки операційної системи Home Assistant
Протоколи для шифрування комунікації
Вбудовані захисти від атак
Принцип найменшого доступу та розділення обов'язків
Огляд операційної системи Home Assistant
Захист від атак XSS
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1.Титульний слайд, 2.Актуальність теми, 3.Завдання кваліфікаційної роботи, 4.Логотип Home Assistant, 5.Галузі застосування Home Assistant, 6.Основні механізми безпеки, 7.Інтерфейс Home Assistant, 8.Управління термостатами, 9.Моніторинг та контроль безпеки, 10. Контроль енергії в розумному будинку, 11-13.Розробка механізму захисту системи від атак XSS

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

7. Дата видачі завдання 29.01.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	Виконано
2.	Підбір джерел про механізми безпеки Home Assistant	20.02 – 27.02	Виконано
3.	Опрацювання джерел в галузі дослідження	28.02 – 16.03	Виконано
4.	Розгортання середовища для проведення аналізу механізмів безпеки	17.03 – 20.03	Виконано
5.	Тестування механізмів безпеки операційної системи	20.03-05.04	Виконано
6.	Оформлення розділу «Галузь застосування та механізми безпеки ОС Home Assistant»	06.03 – 17.04	Виконано
7.	Оформлення розділу «Внутрішня архітектура безпеки в операційній системі Home Assistant»	18.04 – 29.04	Виконано
8.	Оформлення розділу «Практична реалізація»	30.04 – 13.05	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	14.05 – 21.05	Виконано
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	Виконано
11.	Нормоконтроль	06.06 – 12.06	Виконано
12.	Перевірка на плагіат	10.06 – 16.06	Виконано
13.	Попередній захист кваліфікаційної роботи	18.06 – 21.06	Виконано
14.	Захист кваліфікаційної роботи		

Студент _____ Слободян П.П.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Деркач М.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Аналіз механізмів безпеки операційної системи Home Assistant// Кваліфікаційна робота ОР «Бакалавр» //Слободян Поліна Петрівна// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2024 // С. __ , рис. – 21, табл. – 1 , кресл. –__ , додат. – 1.

Ключові слова: аналіз, Home Assistant, розумний будинок, Інтернет речей вразливість, безпека, операційна система.

Кваліфікаційна робота присвячена аналізу механізмів безпеки та реалізації архітектури захисту операційної системи Home Assistant від зовнішніх атак.

У першому розділі кваліфікаційної роботи описані галузі застосування операційної системи, проаналізовано публікації, які стосуються об'єкту дослідження, обрано та обґрунтовано механізми забезпечення безпеки Home Assistant.

У другому розділі кваліфікаційної роботи проаналізовані протоколи, що використовуються для шифрування комунікації, вбудовані захисти від атак та принципи найменшого доступу та розділення ролей.

У третьому розділі кваліфікаційної роботи налаштовано внутрішню архітектуру безпеки Home Assistant та реалізовано механізм захисту від XSS-атак, який дозволяє очищувати HTML-дані, надіслані користувачем, перед відображенням їх у веб-інтерфейсі системи.

ANNOTATION

Analysis of the security mechanisms of the Home Assistant operating system// Thesis of educational level "Bachelor" // Slobodian Polina Petrivna// Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, SB-41 group // Ternopil, 2024 // P. __ , fig. - 21, table. – 1, chair. –__ , added. - 1.

Keywords: analysis, Home Assistant, smart home, Internet of Things vulnerability, security, operating system.

The qualification work is devoted to the analysis of security mechanisms and implementation of the Home Assistant operating system protection architecture against external attacks.

In the first section of the qualification work, the areas of application of the operating system are described, the publications related to the research object are analyzed, the mechanisms for ensuring the security of Home Assistant are selected and substantiated.

In the second section of the qualification work, the protocols used for encryption of communication, built-in defenses against attacks and the principles of least access and separation of roles are analyzed.

In the third section of the qualification work, the internal security architecture of Home Assistant is configured and the protection mechanism against XSS attacks is implemented, which allows to clean the HTML data sent by the user before displaying it in the web interface of the system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОП – операційна система

2FA – двофакторна автентифікація

CSP – Content Security Policy

XSRF – Cross-Site Request Forgery

SCADA – Supervisory Control and Data Acquisition

SSL – Secure Socket Layer

TLS – Transport Layer Security

HTTPS – Hypertext Transfer Protocol Secure

MQTTs – MQTT Secure

QoS – Quality of Service (якість обслуговування)

IoT – Internet of Things (Інтернет речей)

SSH – Secure Shell

PKI – інфраструктура відкритого ключа

CA – незалежний центр сертифікації

ASLR – рандомізація розміщення адресного простору

SHE – структурована обробка винятків

SQLi – SQL-ін’єкція

CSRF – Cross Site Request Forgery

RBAC – Role-Based Access Control

PoLP – принцип найменших привілеїв

ACL – список контролю доступу

ВСТУП

У сучасному світі, який насичений високотехнологічними рішеннями, питання інформаційної безпеки і конфіденційності стали одними з найважливіших аспектів розвитку програмного забезпечення. Одним із найактуальніших напрямків на сьогодні - є захист операційних систем від потенційних кіберзагроз та кіберінцидентів. Однак із збільшенням функціональності з'являється і новий виклик - забезпечення безпеки та конфіденційності користувачів у мережі.

Аналіз механізмів безпеки операційних систем домашніх помічників є актуальним завданням. У цьому контексті вивчення способів захисту від несанкціонованого доступу, зловмисних атак і витоку даних має вирішальне значення для забезпечення стабільності та надійності системи.

У випадку операційної системи Home Assistant, дослідження способів захисту від зловмисних атак, несанкціонованого доступу, та витоків даних є достатньо критичним для забезпечення стабільності та надійності системи. Важливо розуміти обмеження Home Assistant, адже вона може бути складною в налаштуванні, схильною до кіберзагроз, не сумісною з усіма пристроями та потребувати доступу до Інтернету.

Незважаючи на це, Home Assistant залишається потужною та універсальною платформою, яка пропонує користувачам широкий спектр функцій та механізмів безпеки.

1 ГАЛУЗЬ ЗАСТОСУВАННЯ ТА МЕХАНІЗМИ БЕЗПЕКИ HOME ASSISTANT

1.1 Галузь застосування операційної системи Home Assistant

Home Assistant – це безкоштовна платформа з відкритим кодом для автоматизації розумного будинку, яка працює на різних платформах, включаючи мікроконтролери, ПК та інші пристрої [1, 2]. На рисунку 1.1 зображений вигляд веб-сайту операційної системи.

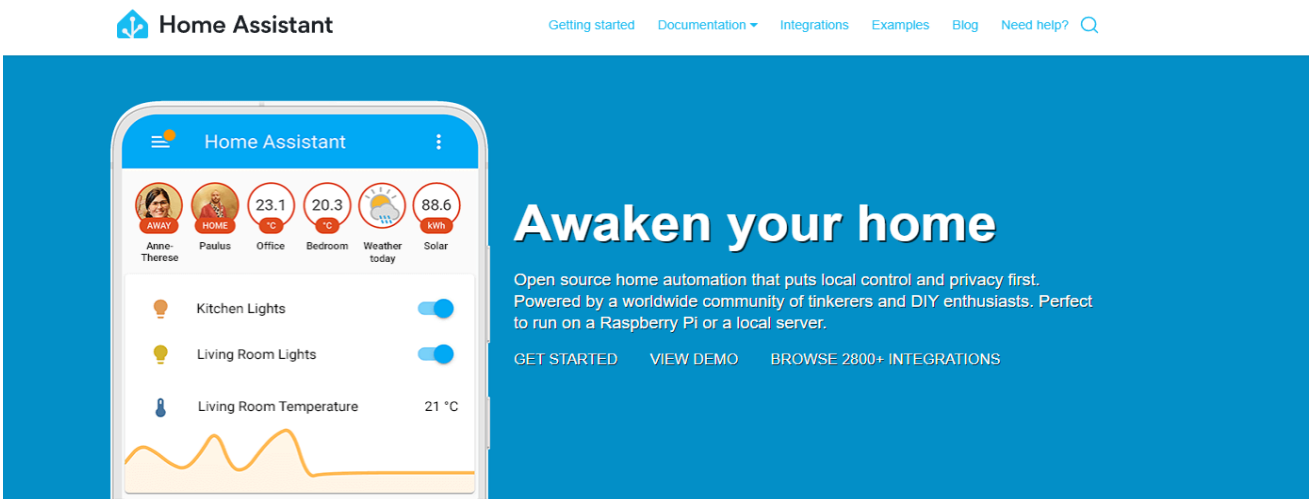

Рисунок 1.1 – Веб-сайт операційної системи Home Assistant

Вона пропонує широкий спектр функцій для автоматизації освітлення, термостатів, камер безпеки, розумних пристроїв та багато іншого. Галузь застосування можна розділити на кілька пунктів:

– автоматизація освітлення, Home Assistant можна використовувати для автоматичного ввімкнення та вимкнення освітлення, регулювання яскравості та створення сцен освітлення;

– управління термостатами, Home Assistant можна використовувати для автоматичного регулювання температури вдома, створення розкладів та віддаленого керування термостатами;

– моніторинг та контроль безпеки, Home Assistant можна використовувати для підключення камер безпеки, датчиків руху та інших пристроїв безпеки для моніторингу вашого будинку та отримання сповіщень у разі небезпеки;

– управління розумними пристроями, Home Assistant можна використовувати для підключення та керування розумними пристроями, такими як телевізори, пральні машини, кавоварки та багато іншого;

– створення голосових команд, Home Assistant можна використовувати для створення голосових команд для керування своїми розумними пристроями за допомогою таких помічників, як Google Assistant та Amazon Alexa;

– інтеграція з іншими системами, Home Assistant можна інтегрувати з іншими системами автоматизації, такими як IFTTT та SmartThings [3-5].

Home Assistant знаходить широке застосування в різних сферах: домашня автоматизація, бізнес, промисловість, державний сектор та освіта. Система пропонує широкий спектр функцій, простоту використання, відкритий код, велику спільноту користувачів, підтримку різних пристроїв та доступні ціни.

Якщо виділити три ключові сфери застосування ОС Home Assistant, якими є домашня автоматизація, комерційне застосування та промислове застосування, то можна знайти кілька сильних та слабких сторін. Сильною стороною є велика та активна спільнота користувачів Home Assistant може надавати підтримку та інформацію про безпеку.

Але у ОС Home Assistant є не лише сильні сторони. Слабкістю системи є, до прикладу, вразливість програмного забезпечення. Як і будь-яка система, Home Assistant може бути вразливою до нових вразливостей програмного забезпечення, які можуть бути виявлені зловмисниками. Також атаки типу "людина в середині" і атаки з використанням соціальної інженерії, де у першому випадку зловмисники можуть перехопити дані, що передаються між Home Assistant та пристроями, якщо вони зможуть отримати доступ до мережі, а у другому – зловмисні користувачі можуть використовувати навіть методи соціальної інженерії, аби змусити користувачів обманом розкрити свої паролі або іншу конфіденційну інформацію. Деякі пристрої, що підключаються до Home Assistant, можуть мати власні вразливості, які можуть бути використані зловмисниками для отримання доступу до системи.

У випадку з комерційним застосуванням сильні сторони досить схожі, але слабкі відрізняються. Зокрема, комерційні приміщення можуть бути більш

привабливими цілями для кібератак, ніж житлові будинки, що робить Home Assistant більш вразливою до атак, а також такі застосування часто потребують більш високого рівня надійності та доступності, ніж житлові, що може бути складніше досягти.

Сильні сторони промислового застосування полягають у повному контролі процесів, де Home Assistant може використовуватися для контролю та автоматизації промислових процесів, що може підвищити ефективність та безпеку. Також система може використовуватися для збору даних про роботу промислового обладнання, що може допомогти у виявленні та вирішенні проблем і для прогнозування можливих проблем з обладнанням, що може допомогти запобігти їх виникненню. Цікавим плюсом є те, що Home Assistant може інтегруватися з існуючими системами SCADA (Supervisory Control and Data Acquisition), що може допомогти централізувати управління та моніторинг промислових процесів. Вагомим мінусом є те, що система не розроблена спеціально для промислових застосувань, тому може не мати необхідних функцій та підтримки і збої в роботі можуть призвести до серйозних проблем з безпекою та дороговартісних пристроїв.

Відповідальне та обґрунтоване використання Home Assistant може допомогти створити безпечні, зручні та ефективні системи автоматизації в різних сферах.

Якщо розглядати приклади домашньої автоматизації, то гарним прикладом такої може бути розумне освітлення, тобто операційна система (далі - ОП) дає можливість вмикати світло у вітальні, при вході до неї, використовуючи датчик руху. Також можна створити сцену освітлення для читання, яка тьмянить освітлення та вмикає настільну лампу. Коли хтось стукає у двері, можна легко дистанційно ввімкнути світло на ганку.

Також функції ОС дозволяють підтримувати комфортну температуру вдома, автоматично регулюючи термостат протягом дня. Дуже зручною можливістю є вмикання обігріву з будь-якої точки міста. ОС дозволяє значно зекономити енергію, не знаходячись вдома, автоматично знижуючи температуру [6].

Користувач ОС Home Assistant може отримувати сповіщення на свій телефон, коли спрацьовує камера безпеки, а також записувати відео з камер безпеки, не знаходячись вдома. Коли людина не вдома, якщо виявляється рух у будинку можна ввімкнути сирену.

Дуже зручною функцією є управління розумними пристроями. Можна ввімкнути кавоварку, коли ви прокидаєтесь вранці або телевізор, коли ви заходите до вітальні. А також створити голосові команди для цих дій. До прикладу ввімкнути освітлення, кажучи «Вмикай світло» або запустити пральну машину, кажучи «Запусти пральну машину».

Дуже хорошим прикладом автоматизації освітлення можуть бути лампи IKEA Tradfri. Home Assistant використовується для автоматичного ввімкнення та вимкнення ламп, а також регулювання їх яскравості. Це може бути корисно для створення атмосфери у кімнаті або заощадити енергію. Для використання знадобиться ОП Home Assistant, лампи IKEA Tradfri і з'єднання Wi-Fi.

Інструкції:

1. Встановіть Home Assistant на свій пристрій.
2. Додайте лампи IKEA Tradfri до Home Assistant.
3. Створіть автоматизацію, яка вмикає/вимикає лампи або регулює їх яскравість за вашими вподобаннями [7].

Також ОП Home Assistant дозволяє налаштувати душ на свій смак і систему вентиляторів, які будуть контролювати вологість.

Коли вологість підвищиться вище встановленого відсоткового значення протягом певного періоду часу, вентилятор увімкнеться. Також є можливість встановити максимальний відсоток вологості, який також увімкне вентилятор. Коли вологість падає нижче заданого значення у відсотках протягом певного періоду часу, система чекає на час затримки, а потім вимикає вентилятор.

Працює з високошвидкісним вентилятором за допомогою «Параметрів швидкості вентилятора». Можна вибрати ступінь вентилятора перед вимкненням (Високий / Низький / ВИМК), або просто вимкнути його (УВІМК. / ВИМК.).

Є можливість встановити «Зимовий режим». Це дозволяє мати різні налаштування для холодних місяців року. Також є можливість увімкнути світло

та скористатися опцією «Керування світлом». Це дозволяє встановити рівень яскравості, колірну температуру в кельвінах і час переходу для світла.

Є можливість використовувати ручний перемикач вентилятора. Це дозволяє вмикати та вимикати вентилятор вручну, і все ще дозволяє запускати автоматичне керування. Також є опція автоматичного вимкнення, щоб він міг працювати як таймер запуску.

Є можливість використовувати «Параметр зв’язку автоматизації». Це вимикає інші засоби автоматизації за допомогою параметра обхідного керування, коли автоматизацію активовано [8].

1.2 Механізми безпеки операційної системи Home Assistant

Основні механізми безпеки, що використовуються в Home Assistant, такі:

- Аутентифікація та авторизація.
- Шифрування та патчі.
- Захист від вразливостей.
- Резервне копіювання та відновлення.
- Захист даних.
- Контроль доступу.
- Моніторинг та журналювання.

Не менш важливим є брандмауер, так як він забезпечує захист системи від несанкціонованого доступу з Інтернету [9-12]. Також, як і кожна система, Home Assistant потребує регулярних оновлень, бо вони надають виправлення відомих вразливостей програмного забезпечення. Хорошою функцією є можливість інтеграції з сторонніми системами безпеки, як-от сигналізація та камери відеоспостереження.

1.2.1 Автентифікація та авторизація

Home Assistant використовує комбінацію методів автентифікації та авторизації для захисту своїх систем та даних, тобто перевірку особистості

користувачів та надання їй доступу до системи і шифрування – захист даних, що передаються між системою та користувачами.

Автентифікація – це процес підтвердження особистості, щоб переконатися, що людина є тим, за кого себе видає. Це часто передбачає введення логіна та пароля, але можуть використовуватися й інші методи, як-от смарт-карти, відбитки пальців тощо [13]. На рисунку 1.2 описана різниця між авторизацією і автентифікацією.

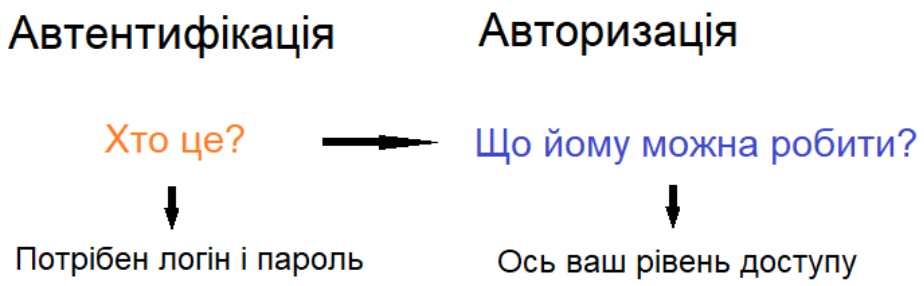

Рисунок 1.2 – Різниця між авторизацією і автентифікацією

Авторизація – це процес перевірки, чи має автентифікований користувач право виконувати певні дії (які часто називають ресурсами). Це зазвичай відбувається шляхом перевірки, чи призначена користувачеві роль, яка надає доступ до цих ресурсів [14].

1.2.2 Захист від вразливостей

Для захисту від вразливостей ОП Home Assistant використовує кілька методів. Як і будь-яка система, що підключається до Інтернету, Home Assistant схильна до кіберзагроз. Найпоширеніші загрози включають вразливості веб-безпеки (можуть дозволити зловмисникам отримати доступ до Home Assistant через Інтернет), вразливості локального виконання коду (можуть дозволити зловмисникам виконати довільний код на Home Assistant), вразливості шифрування (можуть дозволити зловмисникам перехопити або розшифрувати дані) та вразливості конфігурації (можуть бути спричинені неправильною конфігурацією Home Assistant, що робить його більш вразливим до атак).

Home Assistant може бути просканованою на наявність відомих

вразливостей за допомогою спеціальних інструментів. Також ОП використовує різні веб-безпечні практики, такі як Content Security Policy (CSP) та Cross-Site Request Forgery (XSRF) protection, для захисту від поширених веб-атак.

1.2.3 Моніторинг та журналювання

Home Assistant може бути налаштований на моніторинг активності та ведення журналів, що може допомогти виявити підозрілу активність та розслідувати можливі інциденти безпеки [15]. ОС Home Assistant використовує систему журналювання на основі файлів для запису інформації про події, що відбуваються в системі. Записи журналу можуть містити інформацію про дії користувачів, до прикладу вхід та вихід до системи, зміни конфігурації та виконання команд. Також доступна інформація про запуск служб, оновлення, перезавантаження, помилки та події, що генеруються інтегрованими пристроями та службами [16,17].

Home Assistant пропонує декілька рівнів журналювання, що дозволяє контролювати обсяг записуваної інформації:

- DEBUG: записує найдетальнішу інформацію, включаючи дрібні події та налагоджувальні повідомлення.
- INFO: записує інформацію про важливі події та зміни конфігурації.
- WARNING: записує попередження про можливі проблеми.
- ERROR: записує помилки та аварійні завершення роботи.
- CRITICAL: записує критичні помилки, які можуть призвести до збою системи.

Журнали Home Assistant можна переглядати за допомогою веб-інтерфейсу, командного рядка або сторонніх інструментів журналювання, так як ОС пропонує вбудований веб-інтерфейс для перегляду журналів. Також журнали можна переглядати за допомогою команди “*hass.logbook”*.

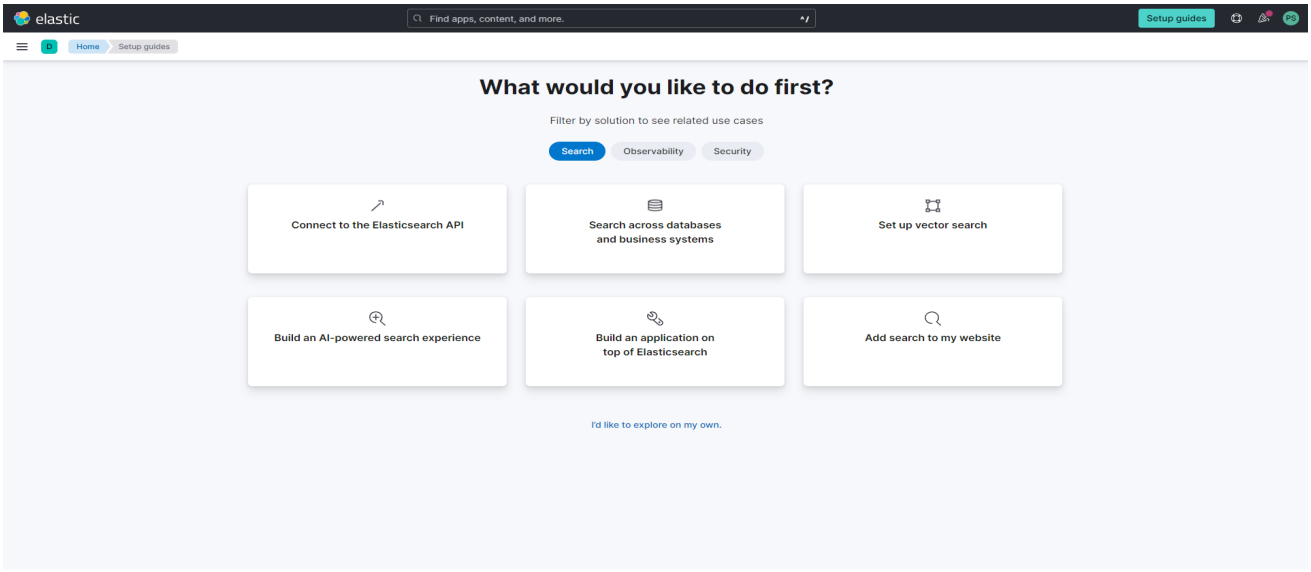

Рисунок 1.3 – Інтерфейс Elasticsearch

Зберігати їх можна локально або на віддаленому сервері. За замовчуванням вони зберігаються у файлі “homeassistant.log” у папці “~/.homeassistant” або на сервері віддаленого журналювання, таких як Elasticsearch або Logstash (див. рисунок 1.3).

1.2.4 Функції аудиту та встановлення оновлень

Функції аудиту Home Assistant включають запис інформації про дії користувачів, такі як вхід до системи, вихід із системи, зміни конфігурації, запис IP-адрес, з яких користувачі отримують доступ до Home Assistant та зберігання історії змін конфігурації. Способі захистити систему є багато, до прикладу встановлення оновлень. Home Assistant регулярно випускає оновлення програмного забезпечення, які виправляють відомі вразливості. Важливо встановити всі доступні оновлення якомога швидше. Використання надійних паролів для Home Assistant та всіх підключених пристроїв. Не можна використовувати один і той самий пароль для кількох облікових записів. І останнє – регулярне резервне копіювання системи дозволить відновити її у разі атаки. Також Home Assistant рекомендує регулярно створювати резервні копії своїх даних, щоб у разі збою системи або кібератаки можна було відновити дані.

1.3 Постановка завдання

Ця кваліфікаційна робота має на меті проаналізувати наявні механізми безпеки операційної системи Home Assistant з метою виявлення їх ефективності та можливих підвищень у рівні захисту. Шляхом вивчення літературних джерел, аналізу відкритих джерел і експериментів планується дослідити існуючі методи та засоби захисту, визначити їхні переваги та недоліки та запропонувати можливі шляхи оптимізації.

Для досягнення поставленої задачі було здійснено:

- аналіз галузей застосування та механізмів безпеки Home Assistant;
- розглянуто внутрішню архітектуру безпеки в операційній системі;
- аналіз протоколів для шифрування комунікації;
- реалізація механізму захисту від XXS-атак.

Також визначено, що операційна система Home Assistant веде детальний облік усіх дій та подій, які відбуваються в системі. Це дозволяє швидко виявляти та реагувати на підозрілу активність і можливі атаки. Аналізуючи журнали, можна підвищити рівень безпеки та усунути вразливості. Home Assistant підтримує регулярне автоматичне оновлення програмного забезпечення, що дозволяє швидко впроваджувати патчі безпеки та нові функції. Це важливо для підтримки високого рівня захисту від нових вразливостей і загроз.

1.4 Висновки до першого розділу

Було досліджено та проаналізовано галузі застосування операційної системи Home Assistant, оцінено та описано механізми безпеки, що використовуються в ОП, у контексті різних галузей застосування. А також визначено сильні та слабкі сторони механізмів безпеки Home Assistant у різних сценаріях використання.

2 ПРОГРАМНІ ЗАХИСТИ ОПЕРАЦІЙНОЇ СИСТЕМІ HOME ASSISTANT

2.1 Вбудовані функції від атак

Home Assistant пропонує ряд вбудованих функцій для захисту системи від атак. Ці функції охоплюють різні аспекти безпеки, від аутентифікації та авторизації до захисту від шкідливого коду та мережевих атак.

2.1.1 Атака переповнення буфера

Переповнення буфера – це помилка програмного кодування або вразливість, якою можуть скористатися хакери для отримання несанкціонованого доступу до корпоративних систем. Це одна з найвідоміших вразливостей безпеки програмного забезпечення, але вона залишається досить поширеною. Це частково тому, що переповнення буфера може відбуватися різними способами, а методи, які використовуються для його запобігання, часто схильні до помилок.

Помилка програмного забезпечення зосереджена на буферах, які є послідовними розділами обчислювальної пам’яті, які тимчасово зберігають дані під час їх передачі між розташуваннями. Також відоме як buffer overrun, переповнення буфера відбувається, коли обсяг даних у буфері перевищує його ємність. Ці додаткові дані переповнюються в сусідні місця пам’яті та пошкоджують або перезаписують дані в цих місцях [18].

Атака переповнення буфера відбувається, коли зловмисник використовує помилки в програмному коді, щоб виконати шкідливі дії і скомпрометувати систему. Це досягається шляхом зміни потоку виконання програми та перезаписування її пам'яті, що дозволяє зловмиснику пошкодити файли або отримати доступ до даних. Такі атаки часто порушують межі буферів, визначені мовою програмування. Здебільшого переповнення буфера виникає через маніпуляції з пам'яттю та неправильні припущення щодо структури або розміру даних.

Вразливість переповнення буфера (CVE-2022-23292) була виявлена в компоненті *mqtt_router* Home Assistant. Ця вразливість дозволяє зловмисному користувачеві виконати довільний код на системі Home Assistant, відправивши спеціально створене MQTT-повідомлення. Також система випустила оновлення програмного забезпечення, яке виправляє цю вразливість. Користувачам рекомендується оновити свої системи Home Assistant до найновішої версії. Окрім оновлення програмного забезпечення, користувачі Home Assistant також повинні вживати таких заходів для захисту своїх систем: використовувати надійні паролі та двофакторну аутентифікацію (2FA), не встановлювати інтеграції та додатки з ненадійних джерел, регулярно оновлювати програмне забезпечення Home Assistant та інтеграцій, а також слідкувати за новинами про безпеку системи та вживати відповідні заходи у відповідь на нові вразливості.

Сучасні операційні системи пропонують захист під час виконання, який забезпечує додатковий захист від переповнення буфера. Сюди входять стандартні функції безпеки, такі як: рандомізація розподілу адресного простору (ASLR). Атаки переповнення буфера зазвичай вимагають інформації про місцезнаходження виконуваного коду. ASLR випадковим чином переміщує положення поля даних, щоб рандомізувати адресний простір, що робить атаки переповнення майже неможливими. Також, запобігання виконанню даних, бо цей метод запобігає виконанню коду атакою в невиконуваній області, позначаючи область пам'яті як виконувану або неможливу. І останнім є захист від перезапису структурованої обробки винятків (SEHOP). Зловмисник може спробувати замінити структуровану обробку винятків (SEH), вбудовану систему, яка керує апаратними та програмними винятками. Для цього потрібно використовувати атаку переповнення стека, щоб перезаписати записи регістрів винятків, що зберігаються в стеку програм. SEHOP не дозволяє шкідливому коду атакувати SEH та використовувати технологію перезапису.

Впровадження заходів безпеки щодо коду розробки та операційних систем недостатньо для захисту систем вашої організації. Якщо виявлено вразливість, пов'язану з переповненням буфера, дуже важливо швидко виправити програмне забезпечення та зробити його доступним для всіх користувачів.

2.1.2 SQL-ін’єкція

SQL-ін’єкція – це техніка впровадження коду, яка може знищити базу даних і є одним із найпоширеніших методів веб-злому. Конкретніше кажучи – це розміщення зловмисного коду в операторах SQL за допомогою введення веб-сторінки.

SQL-ін’єкція зазвичай відбувається, коли сайт просить користувача ввести дані, як-от ім’я користувача/ідентифікатор користувача, і замість імені/ідентифікатора користувач вписує оператор SQL, який ви несвідомо запустите у своїй базі даних. В лістингу 2.1 поданий приклад, який створює оператор SELECT шляхом додавання змінної (txtUserId) до рядка вибору. Змінна отримується з введення користувача (getRequestString).

Лістинг 2.1 – Приклад SQL-ін’єкції

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

Для здійснення атаки SQL-ін'єкції зловмисник спочатку мусить виявити вразливість у веб-сторінці або веб-додатку, де дані користувача використовуються безпосередньо у SQL-запитах. Після цього зловмисник може вставити шкідливий вміст, відомий як корисне навантаження, яке є ключовою частиною атаки. Після того, як цей вміст надісланий, в базі даних виконуються зловмисні SQL-команди.

SQL — це мова запитів, створена для управління даними, які зберігаються у реляційних базах даних. Вона дозволяє здійснювати доступ до даних, їх зміну та видалення. Багато веб-додатків і веб-сайтів використовують бази даних SQL для зберігання всіх даних.

Іноді SQL-ін'єкції можуть бути використані для виконання команд операційної системи. Це означає, що успішна атака може призвести до серйозних наслідків [19].

Цей тип атаки може бути використаний для крадіжки даних, зміни налаштувань або навіть повного компрометування системи Home Assistant а

також зловмисний користувач може змінити налаштування системи, щоб вивести з ладу пристрої, порушити правила автоматизації або навіть отримати доступ до інших систем.

Запобігти вразливості SQL-ін’єкції нелегко. Конкретні методи запобігання залежать від механізму бази даних SQL, підтипу вразливості SQLi і мови програмування. Однак існують певні загальні стратегічні принципи, яких потрібно дотримуватися, щоб зберегти веб-програму в безпеці.

Будь-який ввід користувача Home Assistant, який використовується в SQL-запиті, створює ризик ін’єкції SQL. Тому система обробляє вхідні дані від автентифікованих та/або внутрішніх користувачів так само, як публічні введення. Система використовує бази даних SQLite, але старі технології веб-розробки не мають захисту SQLi. Тому потрібно використовувати останню версію середовища та мови розробки та новітні технології, пов’язані з цим середовищем/мовою. Наприклад, у PHP бажано використовувати PDO замість MySQLi [20].

2.1.3 Міжсайтова підробка запитів

Міжсайтова підробка запитів (CSRF) є атакою, яка змушує автентифікованого користувача веб-програми виконувати небажані дії. Це досягається через соціальну інженерію, наприклад, шляхом надсилання електронних листів або повідомлень у чатах, які змушують користувача виконувати дії за бажанням атакуючого. У випадку, якщо жертва — звичайний користувач, успішна атака CSRF може призвести до виконання запитів на зміну стану, таких як переказ коштів або зміна контактної інформації. В разі, якщо жертвою є адміністратор, CSRF може призвести до компрометації всієї веб-програми [21].

CSRF-атаки можуть мати серйозні наслідки для користувачів Home Assistant, до прикладу зловмисний користувач може отримати контроль над системою, змінюючи налаштування або запускаючи автоматизації. Також,

зловмисник може використовувати систему для здійснення онлайн-покупок або крадіжки особистих даних і розповсюдження шкідливого контенту або спаму.

Home Assistant використовує токен CSRF для кожного запиту, який надсилається на сервер. Цей токен генерується на стороні клієнта та включається в кожен заголовок HTTP. Сервер перевіряє токен CSRF перед виконанням будь-якої дії, щоб переконатися, що запит дійсно надсилається авторизованим користувачем. Тобто, «секретний ключ» (secret), спеціальне значення, яке генерується випадково при авторизації та зберігається в сесії відвідувача. Його знає лише сервер, відвідувачеві його навіть не показують. Зрозуміло, для різних відвідувачів secret буде різним. Потім на основі ключа генерується той самий "токен" (token). Токен робиться так, щоб з одного боку він був відмінний від ключа secret, зокрема може бути багато токенів для одного ключа, з іншого - щоб було легко перевірити по токену, згенерований він на основі даного ключа чи ні.

Також, Home Assistant використовує SameSite cookies для запобігання CSRF-атакам, які ґрунтуються на сторонніх cookie. SameSite cookies обмежують використання cookie лише для запитів, які надсилаються з того ж домену, що й cookie. Атрибут SameSite дозволяє серверам вказувати, чи надсилаються файли cookie з міжсайтовими запитами, тобто сторонні файли cookie. Міжсайтові запити – це запити, у яких сайт (домен, який можна зареєструвати) та/або схема (http або https) не збігаються із сайтом, який зараз відвідує користувач. Це включає запити, надіслані під час натискання посилань на інших сайтах для переходу на ваш сайт, і будь-які запити, надіслані вбудованим стороннім вмістом.

SameSite допомагає запобігти витоку інформації, зберігаючи конфіденційність користувачів і забезпечуючи певний захист від атак підробки міжсайтових запитів. Він приймає три можливі значення: Strict, Lax і None [22]. Strict означає, що браузер надсилає файли cookie лише у відповідь на запити, що надходять із сайту походження файлів cookie. Це слід використовувати, якщо є файли cookie, пов’язані з функціями, які завжди будуть за початковою навігацією, як-от автентифікація або зберігання інформації про кошик для покупок. Lax схожий, за винятком того, що браузер також надсилає файл cookie,

коли користувач переходить на сайт походження файлу cookie (навіть якщо користувач переходить з іншого сайту). None вказує, що файли cookie надсилаються як на вихідні, так і на міжсайтові запити. Це корисно, якщо потрібно аби файли cookie надсилались разом із запитами, зробленими з вмісту третіх сторін, вбудованого в інші сайти.

І останнє, система Home Assistant перевіряє HTTP Referer header кожного запиту, щоб переконатися, що запит надсилається з очікуваного джерела. Заголовок Referer HTTP-запиту містить абсолютну або часткову адресу запитуваного ресурсу і дозволяє серверу визначити, які сторінки, що посилаються на нього, відвідували користувачі і де використовувався запитуваний ресурс. Ці дані можуть бути використані для аналізу, ведення журналів, оптимізації кешування тощо.

2.2 Принцип найменшого доступу та розділення обов’язків

Принцип найменшого доступу (Principle of Least Privilege) та розділення обов'язків (Role-Based Access Control) є важливими принципами безпеки, які можуть допомогти захистити Home Assistant від несанкціонованого доступу та атак.

Принцип найменших привілеїв (PoLP) відноситься до концепції інформаційної безпеки, згідно з якою користувачеві надаються мінімальні рівні доступу або дозволи, необхідні для виконання його/її службових функцій. Це широко вважається найкращою практикою кібербезпеки та є фундаментальним кроком у захисті привілейованого доступу до цінних даних і активів. Найменший привілей виходить за межі доступу людини.

Модель може бути застосована до програм, систем або підключених пристроїв, яким потрібні привілеї або дозволи для виконання необхідного завдання. Застосування найменших привілеїв гарантує, що нелюдський інструмент має необхідний доступ – і нічого більше.

Для ефективного застосування мінімальних привілеїв потрібен спосіб централізованого керування та захисту привілейованих облікових даних, а також

гнучких елементів керування, які можуть збалансувати вимоги до кібербезпеки та відповідності з операційними потребами та потребами кінцевих користувачів [23]. PoLP допомагає обмежити потенційний збиток, який може бути завданий, якщо зловмисний користувач отримає доступ до системи.

Завдяки принципу найменших привілеїв користувачі не можуть змінювати налаштування, до яких вони не мають доступу, або використовувати функції, які їм не потрібні. Це робить Home Assistant більш безпечною та надійною платформою. Home Assistant пропонує ряд функцій, які допомагають реалізувати PoLP, до прикладу адміністратори системи можуть призначати користувачам різні ролі з різними рівнями доступу. Це гарантує, що користувачі мають доступ лише до тих функцій та налаштувань, які їм необхідні.

Home Assistant підтримує список контролю доступу (ACL), які дозволяють адміністраторам детально контролювати, хто має доступ до яких ресурсів. Список правил ACL – це один з основних способів керування безпекою в комп'ютерних мережах та системах. Існує чотири різні типи ACL – стандартний, розширений, динамічний і рефлексивний.

– Стандартний ACL: стандартний ACL зосереджується на адресі джерела. Цей тип списку контролю доступу в кібербезпеці враховує лише джерело користувача або системи, що запитує. Це найпростіша форма ACL і, отже, не може забезпечити безпеку найвищої якості.

– Розширений ACL: трохи складніший за стандартний ACL, розширений ACL дозволяє блокувати джерело та призначення як для одного хоста, так і для цілої мережі. Крім того, також можна фільтрувати трафік на основі інформації про протокол за допомогою розширеного списку доступу.

– Динамічний ACL: вимагає спеціальної автентифікації, динамічний ACL фактично використовує розширені ACL. Їх можна використовувати для певних часових рамок і часто називають «замок і ключ».

– Рефлексивні ACL: використання інформації сеансу верхнього рівня для фільтрації трафіку, рефлексивні ACL також відомі як ACL IP-сесії. Працюючи в межах певного сеансу, цей вид запису видаляється після завершення сеансу [24].

Контроль доступу на основі ролей (RBAC) – це метод управління доступом, який використовується для надання користувачам доступу до ресурсів на основі їхніх ролей у системі. Цей метод відноситься до ідеї призначення дозволів користувачам на основі їх ролі в організації. Він пропонує простий керований підхід до керування доступом, який менш схильний до помилок, ніж призначення дозволів користувачам окремо.

Home Assistant пропонує вбудовану рольову систему, яка дозволяє адміністраторам створювати різні ролі з різними рівнями доступу. Рівні поділяються на адміністратора, користувача і гостя.

Адміністратор має найвищий рівень доступу і може змінювати будь-які налаштування, створювати власні ролі з настроюваними дозволами та виконувати будь-які дії.

Користувач може лише переглядати інформацію і виконувати базові дії, до прикладу вимикати або вимикати освітлення вдома.

Гість має найменший рівень доступу. Він може переглядати лише обмежену інформацію і не може виконувати жодних дій.

RBAC допомагає обмежити доступ до конфіденційних даних та чутливих налаштувань і робить простішим управлінням дозволами користувачів та зменшує ризик помилок. А також, дозволяє створювати настроюванні ролі, які відповідають специфічним потребам користувачів.

Деякі позначення в інструменті RBAC можуть включати:

– Обсяг ролі керування – обмежує, якими об’єктами дозволено керувати групі ролей.

– Рольова група керування – ви можете додавати та видаляти учасників.

– Роль керування – це типи завдань, які може виконувати конкретна рольова група.

– Призначення ролі керування – це пов’язує роль із групою ролей [25].

Додавання користувача до групи ролей надає йому доступ до всіх ролей у цій групі. Якщо їх видалити, доступ буде обмежено. Крім того, якщо потрібен тимчасовий доступ до певних даних або програм, користувачів можна призначити до кількох груп і видалити їх після завершення проєкту.

2.3 Протоколи для шифрування комунікації

Home Assistant використовує декілька протоколів шифрування для захисту конфіденційності та цілісності даних, що передаються між системою та зовнішніми джерелами інформації. В цьому розділі буде розглянуто найбільш поширені.

2.3.1 Secure Socket Layer

SSL розшифровується як Secure Socket Layer або Рівень Захищених Сокетів. Це попередник TLS, абревіатура від Transport Layer Security (Протокол захисту транспортного рівня), який є криптографічним протоколом, що забезпечує безпечну передачу даних між вузлами комп'ютерної мережі. Схема роботи протоколу зображена на рисунку 2.1 [26].

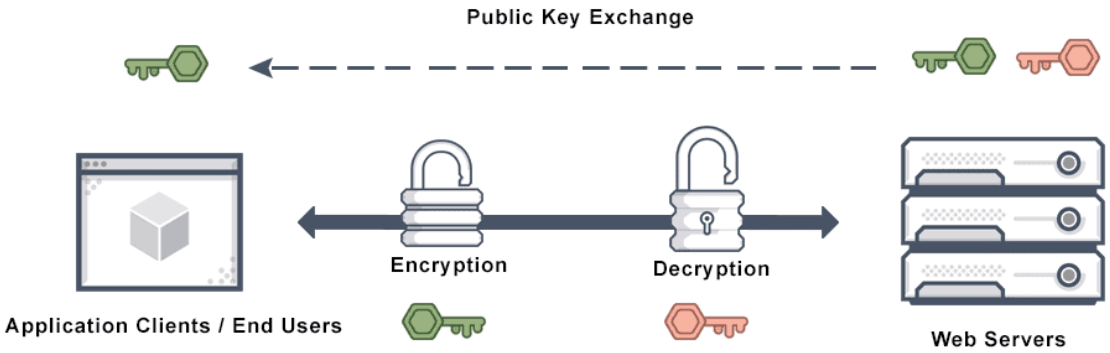

Рисунок 2.1 – Схема роботи SSL

TLS складається з 2 фаз чи 2 протоколів. Перший - протокол рукостискання. На цій фазі клієнт та сервер будуть узгоджувати версію протоколу, вибирати криптографічний алгоритм чи наборів шифрів, автентифікувати один одного за допомогою асиметричної криптографії та визначати загальний секретний ключ, який використовуватиметься для симетричного шифрування на наступній фазі. Таким чином, основна мета рукостискання – автентифікація та обмін ключами. Друга фаза – протокол запису. На цій фазі усі вихідні повідомлення будуть зашифровані за допомогою

загального секретного ключа, встановленого під час рукостискання. Потім зашифровані повідомлення надсилаються іншій стороні. Їх перевіряють, щоб побачити, чи виникли якісь зміни під час передачі чи ні. Якщо ні, повідомлення будуть дешифровані з використанням того ж симетричного секретного ключа.

Таким чином, ми досягнемо як конфіденційності, так і цілісності в цьому протоколі запису, і оскільки обсяг зашифрованих даних на цьому етапі великий, його часто називають шифруванням великих обсягів даних.

Для того, щоб системи могли перевіряти особистість та згодом встановлювати зашифроване мережеве з'єднання з іншою системою за допомогою SSL/TLS потрібен сертифікат. Сертифікати використовуються у рамках криптографічної системи, відомої як інфраструктура відкритого ключа (PKI). PKI дає одній стороні можливість встановлювати справжність іншої сторони за допомогою сертифікатів (за умови, що обидві сторони довіряють третій стороні, відомій як центр сертифікації). Таким чином, сертифікати SSL/TLS діють як цифрові посвідчення особи для захисту мережевих підключень та встановлення автентичності веб-сайтів в Інтернеті, а також ресурсів у приватних мережах.

2.3.2 Hypertext Transfer Protocol Secure

Наступним протоколом є HTTPS – це стандартний протокол для безпечного зв'язку між веб-серверами та клієнтами. Він використовує шифрування TLS для захисту даних від перехоплення та підміни. Home Assistant використовує HTTPS для зв'язку з багатьма зовнішніми сервісами, такими як хмарні служби, API та веб-сайти [27]. Протокол HTTP – це базова технологія, що забезпечує мережевий зв'язок. Як випливає з назви, захищений протокол передачі гіпертексту (HTTPS) є безпечнішою версією або розширенням HTTP (див. рисунок 2.2). При використанні HTTPS браузер і сервер встановлюють безпечне зашифроване з'єднання перед передачею даних.

HTTP передає незашифровані дані, що означає, що інформація, надіслана з браузера, може бути перехоплена та прочитана третіми особами. Цей процес не

є ідеальним, тому він був розширений до HTTPS, щоб підвищити рівень безпеки взаємодії. HTTPS поєднує запити HTTP та відповіді з технологіями SSL і TLS.

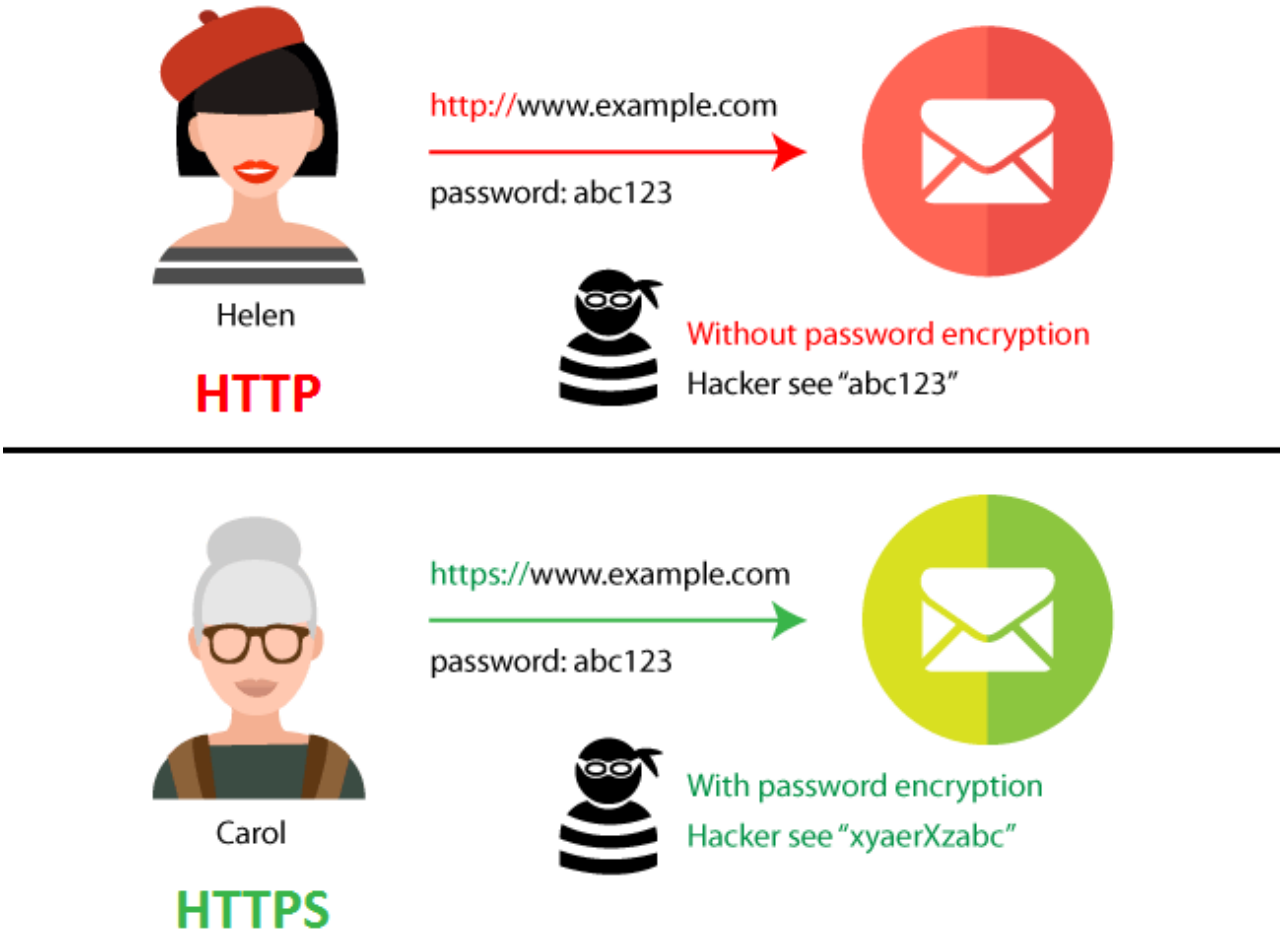

Рисунок 2.2 – Різниця між HTTP і HTTPS

Веб-сайти HTTPS повинні отримати сертифікат SSL/TLS від незалежного центру сертифікації (CA). Ці веб-сайти надсилають сертифікат браузеру, а потім обмінюються даними для встановлення довіри. Також SSL-сертифікат містить криптографічну інформацію, тому сервер та веб-браузери можуть обмінюватися зашифрованими даними. Процес працює в такий спосіб.

Для початку, користувач відкриває веб-сайт HTTPS, ввівши URL-адресу https:// в адресному рядку браузера. Браузер намагається перевірити автентифікацію сайту, запросивши SSL-сертифікат сервера. У відповідь сервер надсилає сертифікат SSL, який містить відкритий ключ. Сертифікат SSL веб-сайту підтверджує особистість сервера. Як тільки браузер задоволений, він

використовує відкритий ключ для шифрування та надсилання повідомлення, що містить секретний ключ сеансу. Веб-сервер використовує свій закритий ключ для розшифрування повідомлення та отримання ключа сеансу. Потім він шифрує сеансовий ключ і відправляє повідомлення, що підтверджує, в браузер. Тепер і браузер, і веб-сервер переходять на використання одного й того ж сеансового ключа для безпечного обміну повідомленнями.

2.3.3 Message Queuing Telemetry Transport Secured

MQTTs – це захищена TLS версія протоколу MQTT. MQTTs означає Message Queuing Telemetry Transport Secured. Він зазвичай використовується для зв'язку між вбудованими системами або пристроями Інтернету речей. Що стосується MQTT, MQTTs може працювати поверх TCP/IP, це дозволяє користувачам налаштовувати бажану якість обслуговування (QoS), щоб забезпечити доставку даних від одного клієнта до іншого [28]. Принцип роботи MQTT розглянуто на рисунку 2.3.

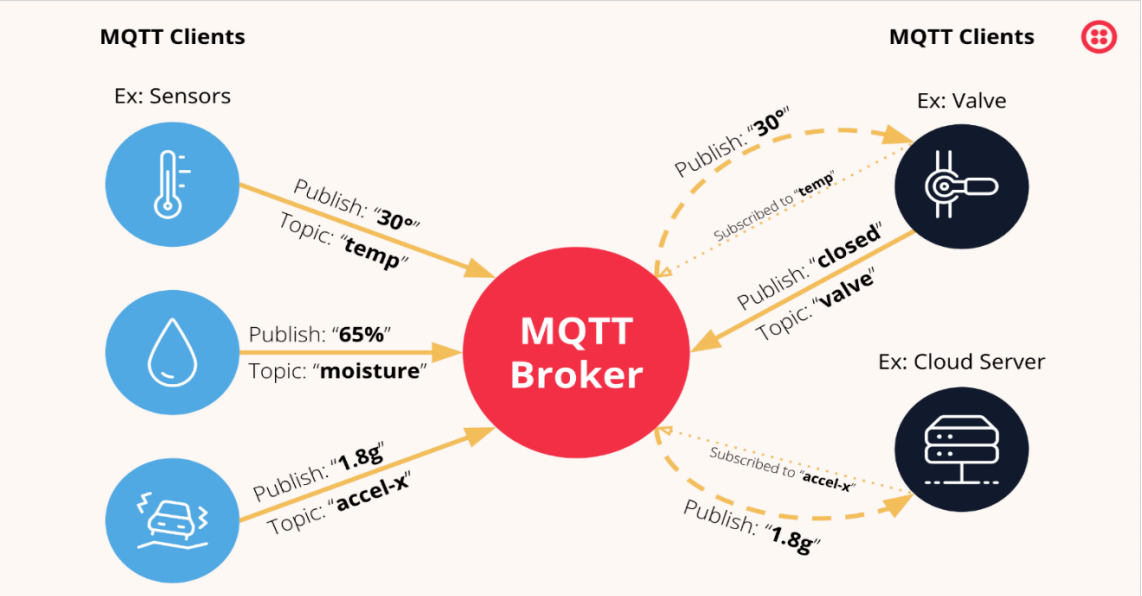

Рисунок 2.3 – Принцип роботи MQTT

На жаль, додатковий захист від MQTT до MQTTS вимагає більше трафіку та швидше розряджає акумулятори пристрою. Використовуючи з'єднувач Onomondo TLS, ми можемо змусити пристрій використовувати MQTT, який

Onomondo перетворює на MQTTS у базовій мережі, використовуючи менше трафіку та зменшуючи енергоспоживання.

У цьому прикладі використовується модуль MQTT для Node.js, але ви можете замінити його будь-якою іншою реалізацією мови програмування. Також використовується test.mosquitto.org як тестовий сервер MQTT. У лістингу 2.2 показано базовий приклад використання MQTTS для підключення:

Лістинг 2.2 – Приклад підключення протоколу MQTTS

```
const fs = require('fs')
const mqtt = require('mqtt')
const ca = fs.readFileSync('./mosquitto.org.crt').toString()
const client = mqtt.connect('mqtts://test.mosquitto.org:8883', { ca })
client.on('message', (topic, payload) =>
   console.log(`[messaged            received]            topic=${topic}
payload=${payload.toString()}`)
)
client.on('connect', () => {
   console.log('[connected]')
   client.subscribe('helpomondo/testing', () =>
      client.publish('helpomondo/testing', 'Hello from device')
   )
})
```

Для того, щоб захопити пакети що йдуть до/з пристрою використовувався Traffic Monitor Onomondo, але також можна використовувати Wireshark. Коли цей маленький приклад запускався, він використовував 3,6 КБ. Більшість з обсягу використовується для встановлення безпечного з’єднання.

2.3.4 Secure Shell

SSH є шифрованим мережевим протоколом, який забезпечує безпечний віддалений доступ та виконання команд через командний рядок в незахищених мережах.

Сучасні фахівці використовують цей стандарт для передачі файлів, виконання команд (віддалено) і управління мережевою інфраструктурою. SSH також часто називають “безпечною оболонкою”. Це тому, що протокол надає безпечний, а саме автентифікований і зашифрований, інтерфейс командного

рядка. За допомогою SSH можна виконувати інші дії, такі як надсилання файлів, запуск програм або зміна налаштувань на іншому комп’ютері. Архітектура протоколу зображена на рисунку 2.4.

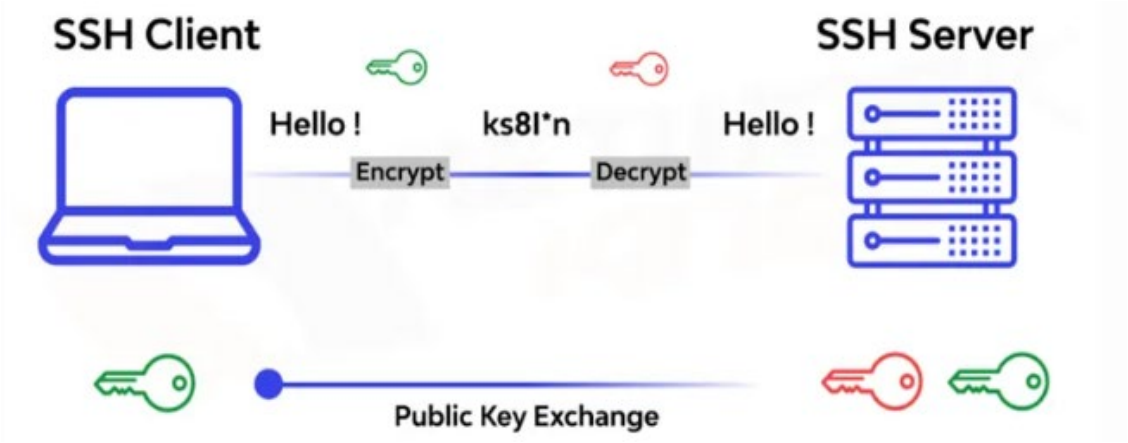

Рисунок 2.4 – Архітектура SSH

Secure Shell – це правила, які описують безпечну взаємодію між комп'ютерами. Він включає три основних компоненти: транспортний рівень, рівень автентифікації користувача та рівень з'єднання. Транспортний рівень забезпечує довіру між комп'ютерами, запобігаючи шпигунству та підробці повідомлень. Розділ користувача дозволяє лише потрібній людині отримати доступ до іншого комп'ютера. Канальний рівень дозволяє комп'ютеру виконувати різні дії одночасно, зокрема, вводити команди, відправляти файли або відкривати програми. Протокол SSH забезпечує безпечний спосіб надсилання та отримання даних через незахищені мережі, такі як Інтернет. Він захищає від різних атак, таких як підслуховування, підробка документів, фальсифікація документів і атаки типу “людина посередині”. SSH також дозволяє виконувати різні завдання віддалено, не розкриваючи паролі або дані хакерам або зловмисникам. Він передає будь-який тип даних через зашифрований канал, якщо клієнт і сервер підтримують один і той же протокол. Наприклад, SSH за допомогою SCP або SFTP може передавати двійкові, текстові, аудіо, відео та графічні файли. Він також використовує переадресацію портів і тунелювання для передачі інших типів даних, таких як графіка, порти TCP/IP і сокети. Однією з найкорисніших функцій SSH є можливість створення

тунельних і проксі-серверів, що допомагає обходити брандмауери, отримувати доступ до обмежених ресурсів, шифрувати небезпечні протоколи і приховувати мережеву активність [29].

2.4 Висновки до другого розділу

У другому розділі описано основні програмні захисти, які забезпечують стійкість системи від різних загроз та захищають дані.

Як загальна платформа для автоматизації домашніх пристроїв, Home Assistant надає користувачам широкий спектр інструментів безпеки: Home Assistant підтримує багатофакторну автентифікацію (MFA), яка забезпечує додатковий надійний рівень захисту облікових записів користувачів. Це запобігає несанкціонованому доступу до системи в разі компрометації облікових даних користувача. Всі дані, що передаються між компонентами Home Assistant, можуть бути зашифровані за допомогою протоколу SSL/TLS. Це захищає інформацію від перехоплення і несанкціонованого доступу під час передачі.

Home Assistant дозволяє встановлювати права доступу для різних користувачів і пристроїв, підключених до системи. Це дозволяє контролювати, хто має доступ до яких ресурсів, і запобігає можливим загрозам з боку внутрішніх користувачів. Щоб знизити ризик порушень системи, рекомендується використовувати окремі мережі для IoT-пристроїв і основного пристрою користувача. Home Assistant підтримує конфігурації, які можуть сегментувати мережу і забезпечувати додатковий рівень захисту. Ці компоненти складають основу внутрішньої архітектури безпеки Home Assistant для надійного захисту даних і системи в цілому.

3 ВНУТРІШНЯ АРХІТЕКТУРА БЕЗПЕКИ HOME ASISSTANT

3.1 Налаштування операційної системи Home Assistant

Для того, щоб спробувати систему Home Assistant, офіційний сайт надає демонстраційний варіант, а також обладнання для легкого і бюджетного початку користування. На веб-сторінці також є чотири варіанти складності для інсталювання системи. Сайт описує які навички та обладнання потрібні для користування, а також дає посилання на придбання засобів та покрокову інструкцію.

Таблиця 3.1 – функції Home Assistant і їх доступність в системах

	Home Assistant Operating System	Home Assistant Container	Home Assistant Core	Home Assistant Supervised
Automations	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	-	✓
Supervisor	✓	-	-	✓
Add-ons	✓	-	-	✓
Backups	✓	✓	✓	✓
Managed Restore	✓	-	-	✓
Managed OS	✓	-	-	-

Перший – легкий варіант, це завантажити Home Assistant на мікрокомп’ютер Raspberry Pi. Наступний спосіб – для досвідчених користувачів, де для встановлення користувачу потрібен елемент Home Assistant Yellow. Наступними є два складних варіантів встановлення, де пропонується встановлення на інше обладнання. Першим з них є – встановлення на Odroid пристроях, а другим – на X86-64. У цьому випадку посилання на купівлю обладнання вже відсутнє. Останній – експертний рівень використання. Він підходить для просунутих користувачів, у яких є специфічні потреби. Цей

варіант пропонує використовувати Home Assistant як віртуальне середовище, або поверх існуючої операційної системи. Але в такому випадку деякі функції операційної системи можуть бути недоступні.

Також сайт пропонує кілька варіантів встановлення на різних ОП, зокрема Linux, MACOS, Windows (див. рисунок 3.1) та інших.

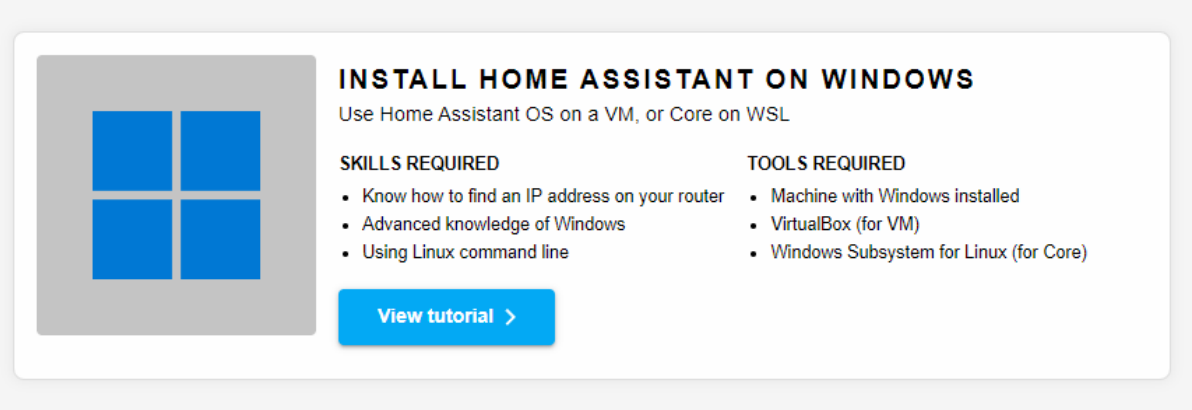

Рисунок 3.1 – Встановлення Home Assistant на Windows

Після натискання на кнопку “View tutorial” відкривається покрокова інструкція по встановленню. Для інсталяції буду використовувати Oracle Virtualbox (див. рисунок 3.2). Завантажити та встановити його можна з офіційного веб-сайту Oracle VirtualBox (https://www.virtualbox.org/wiki/Downloads).

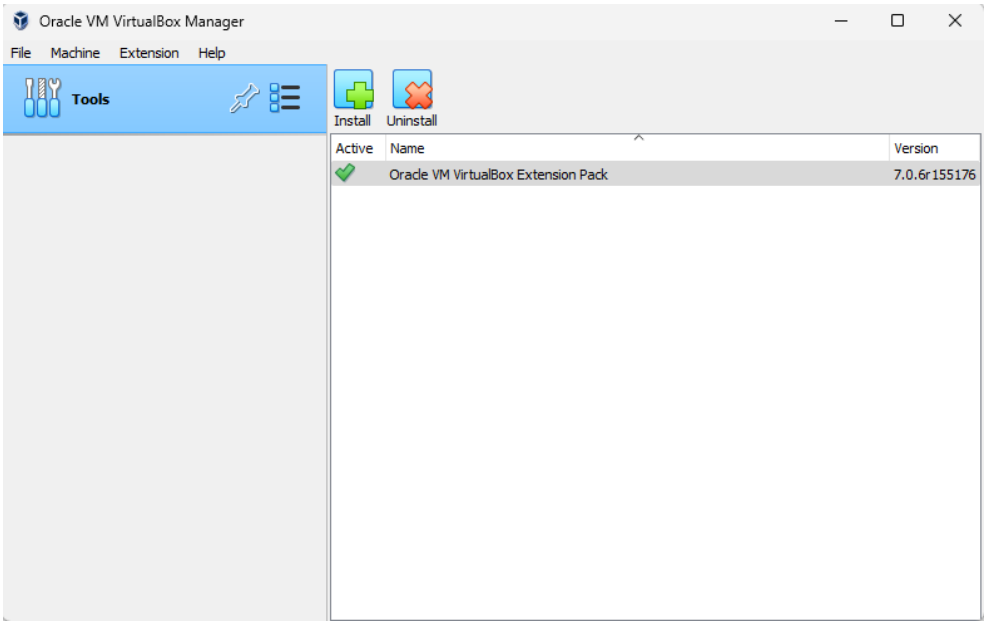

Рисунок 3.2 – Інсталяція віртуальної машини

Для цього треба відкрити VirtualBox і натиснути опцію "New", щоб створити нову віртуальну машину. Далі треба налаштувати параметри віртуальної машини, такі як ім'я, тип і версія ОП. Потім встановити кількість оперативної пам'яті та кількість процесорів, які потрібно призначити віртуальній машині (див. рисунок 3.3).

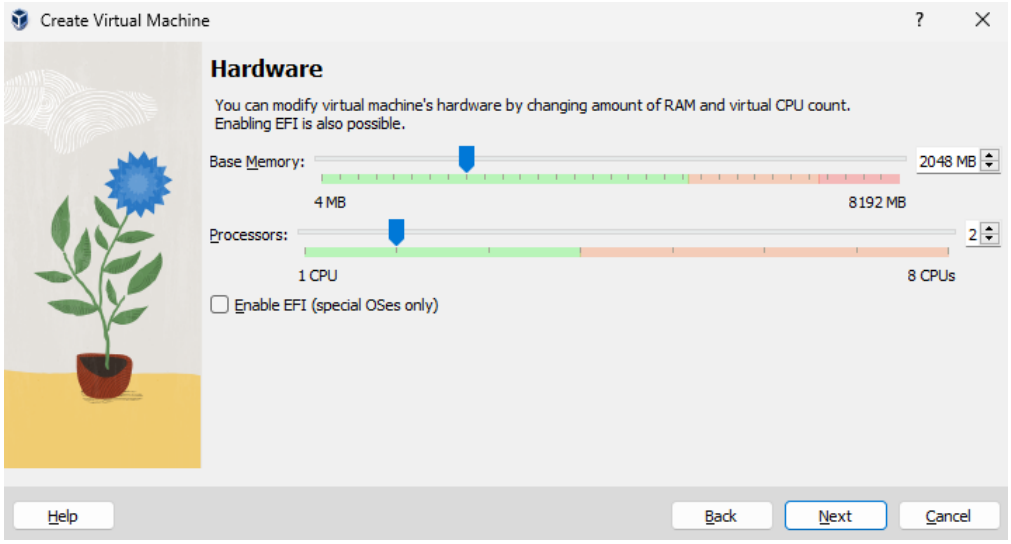

Рисунок 3.3 – Налаштування параметрів віртуальної машини

Для налаштування образу віртуального диска обираємо опцію "Create a virtual hard disk now", тобто створюємо віртуальний жорсткий диск. Залишається обрати тип віртуального диска, розмір диска та зберегти налаштування (див. рисунок 3.4).

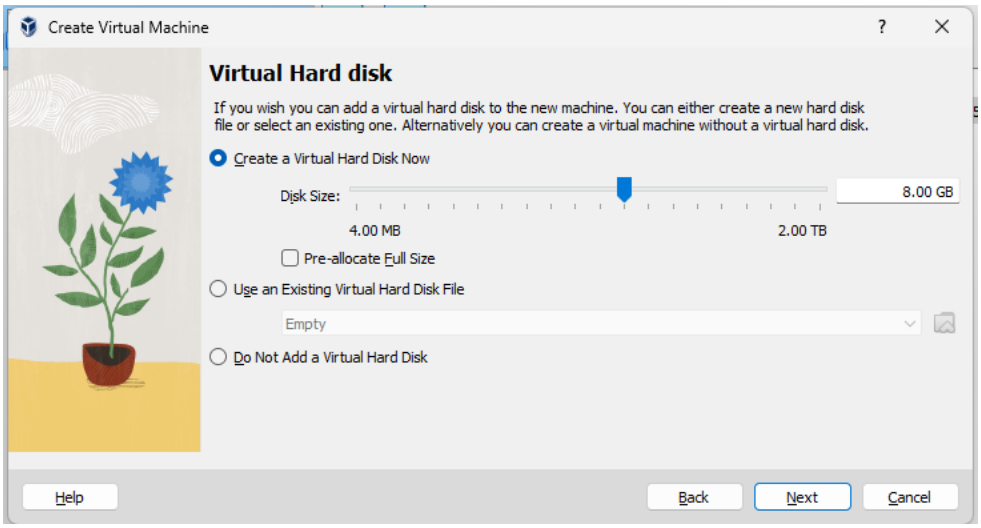

Рисунок 3.4 – Налаштування параметрів віртуальної машини

Далі налаштовано DHCP Client, тобто функцію, яка дозволяє пристрою отримати конфігураційні параметри від DHCP-сервера, зокрема IP-адресу.

DHCP (Dynamic Host Configuration Protocol) є протоколом, який дозволяє автоматизувати процес призначення IP-адреси та інших параметрів мережі для пристроїв у комп'ютерних мережах, а саме:

– Отримання IP-адреси: DHCP Client дозволяє пристрою автоматично отримати IP-адресу від DHCP-сервера.

– Окрім IP-адреси, DHCP Client може отримати інші конфігураційні параметри, такі як маска підмережі, шлюз, DNS-сервер і т. д.

– Автоматичне оновлення конфігурації: DHCP Client періодично оновлює свою конфігурацію, спробуючи зберегти актуальні дані від DHCP-сервера.

Налаштування можна зробити у вкладці Ip/dhcp-client (див. рисунок 3.5), додаємо інтерфейс та залишаємо правила за замовченням:

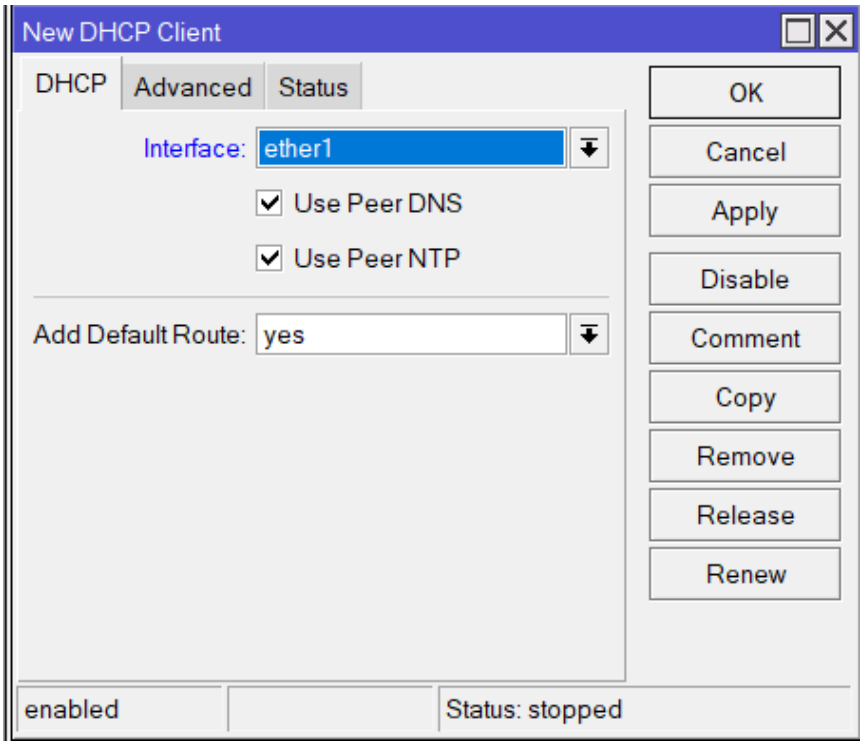

Рисунок 3.5 – Створення DHCP Client

Після налаштування віртуальна машина отримає IP-адресу від основного маршрутизатора (див. рисунок 3.6).

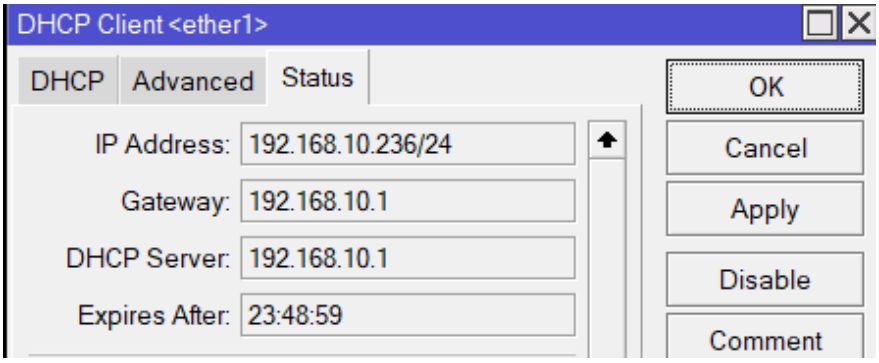

Рисунок 3.6 – Налаштування DHCP Client

Після всіх необхідних налаштувань Home Assistant стає доступним за вказаною IP-адресою, як показано на рисунку 3.7.

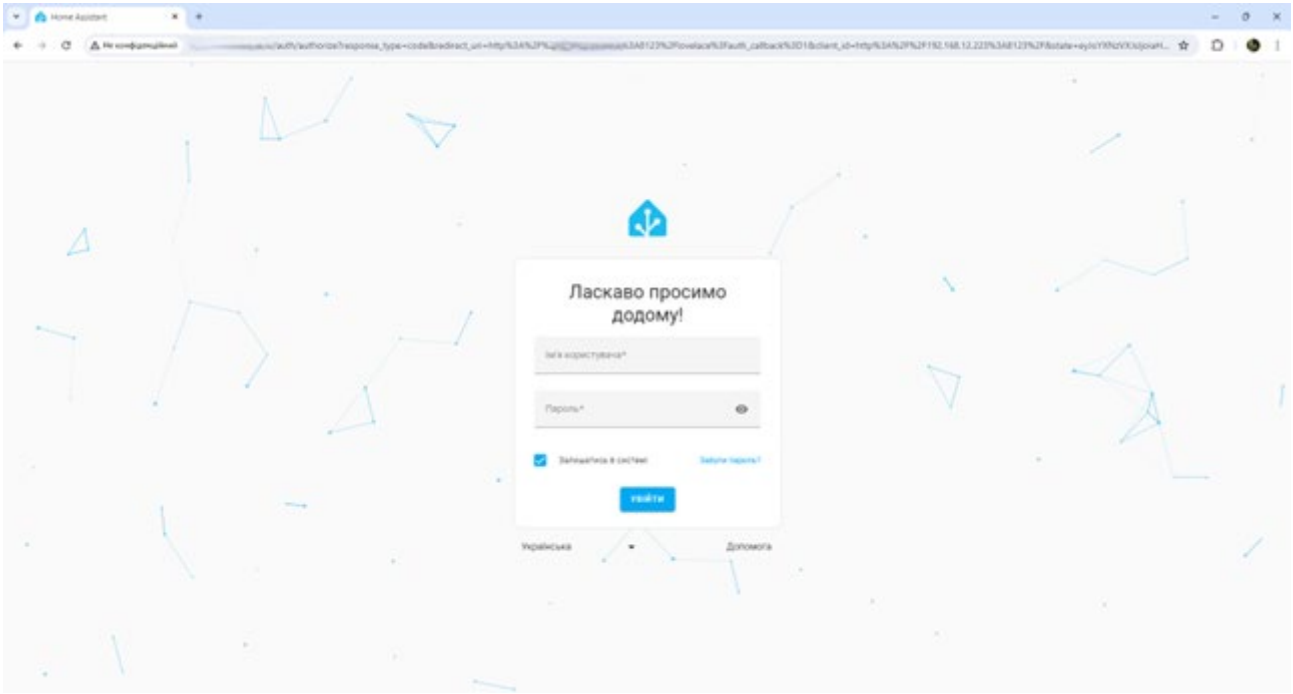

Рисунок 3.7 - Інтерфейс для авторизації в Home Assistant

Значною перевагою використання системи у домашній автоматизації є контроль доступу, бо Home Assistant використовує механізми аутентифікації та авторизації для контролю доступу до системи, що дозволяє обмежувати доступ лише авторизованим користувачам. Також плюсом є те, що дані, що передаються між системою та пристроями, зазвичай шифруються, захищаючи їх від несанкціонованого перехоплення.

Home Assistant регулярно випускає оновлення програмного забезпечення, які виправляють відомі вразливості та покращують безпеку.

ОП Home Assistant підтримує різні методи автентифікації, включаючи логін та пароль, що є найбільш поширеним методом де користувач вводить дані для доступу до системи. Також двофакторну автентифікацію (далі - 2FA), що додає додатковий рівень безпеки, вимагаючи від користувача ввести код підтвердження, який надсилається на його телефон або електронну пошту, окрім пароля. І останнє – інтеграція з сторонніми системами автентифікації, так як Home Assistant може бути інтегрований з сторонніми системами аутентифікації, до прикладу Google Authenticator або Microsoft Azure Active Directory.

Після успішної аутентифікації Home Assistant використовує систему на основі ролей для надання користувачам різних рівнів доступу до системи. Це означає, що користувачі можуть мати доступ лише до певних функцій та даних, залежно від їхньої ролі.

Також ОП використовує шифрування для захисту даних як у стані спокою, так і під час передачі. Дані, що зберігаються в Home Assistant, шифруються за допомогою AES-256, що є стійким алгоритмом шифрування. ОП може використовувати HTTPS для шифрування трафіку між системою та користувачем, захищаючи дані від перехоплення та підслуховування.

Home Assistant регулярно оновлюється для виправлення помилок безпеки і покращення загальної безпеки системи. Користувачам рекомендується постійно оновлювати свою систему до останньої версії, аби захист був максимальним.

Home Assistant пропонує досить комфортний і зручний інтерфейс. Візуальний дизайн веб-інтерфейсу ОП Home Assistant досить сучасний, що робить інформацію легкою для розуміння. Інтерфейс складається з панелей і карток, які можна налаштувати відповідно до потреб користувача. Це дозволяє користувачам створювати власні інформаційні панелі з необхідними їм компонентами, значно спрощуючи управління своїм розумним будинком.

Головне меню веб-інтерфейсу ОП Home Assistant розташоване збоку, забезпечуючи швидкий доступ до основних функцій і налаштувань системи.

Користувачі можуть легко перемикатися між оглядом стану пристроїв, автоматизацією, сценаріями, журналами та налаштуваннями. Це робить навігацію зручною та зрозумілою навіть для новачків.

Одним з ключових аспектів простоти використання веб-інтерфейсу Home Assistant є можливість налаштування автоматизації та сценаріїв без необхідності програмування. Інтерфейс дозволяє користувачам створювати правила автоматизації за допомогою простого редактора, який дозволяє вибирати умови та дії зі спадного списку. Це робить процес налаштування автоматизації швидким і простим, що особливо корисно для тих, хто не має технічного досвіду.

Важливим елементом зручності є багатомовність веб-інтерфейсу Home Assistant. Користувачі можуть вибирати з широкого спектру мов, що дозволяє користувачам з різних країн і регіонів отримати доступ до системи.

Варто також відзначити, що система може бути інтегрована з широким спектром пристроїв і сервісів. Інтерфейс Home Assistant підтримує різноманітні інтеграції, які можна легко додавати та налаштовувати за допомогою зручного майстра налаштувань (див. рисунок 3.8). Це дозволяє користувачам легко підключати нові пристрої та сервіси до системи автоматизації [30-31].

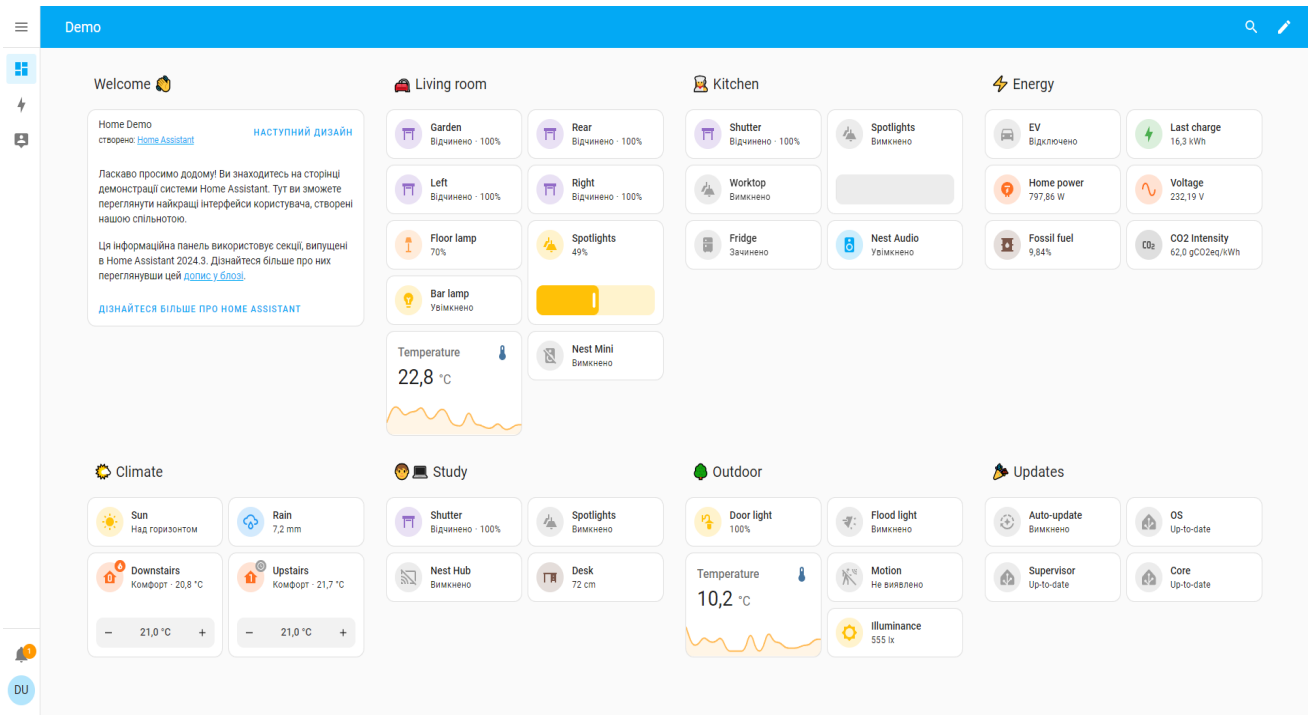

Рисунок 3.8 – Інтерфейс Home Assistant

Розглянемо більш детально меню операційної системи, що складається з вкладок “Огляд”, “Енергія”, “Мапа”, “Сповіщення” та вкладка для налаштування системи.

Перша вкладка “Огляд” демонструє багато віджетів для керування розумним будинком. Сторінка розділена на групи віджетів по кімнатам, в яких розташовані відповідні елементи. Наприклад, можна регулювати у відсотках стан відкриття/закриття жалюзі на вікні. Також, окрім вікон, елементами є лампи у кімнаті. До прикладу, за допомогою ОС Home Assistant можна регулювати освітлення з спеціальними повзунками і обрати ефект освітлення і колір (теплий/холодний). Окрім цього, система дозволяє обрати атрибути освітлення, тобто воно може бути статичним і динамічним. Однією з важливих функцій є налаштування температури в домівці. Home Assistant дозволяє дивитись на статистику температури вдома погодинно, а також показує температуру в даний момент (див. рисунок 3.9). За допомогою ОС Home Assistant можна перевіряти коли був зачинений та відчинений холодильник. Також за допомогою ОС Home Assistant можна вмикати та вимикати аудіопристрої, до прикладу колонки, регулювати звук, ставити аудіо на паузу і обирати медіа з списку.

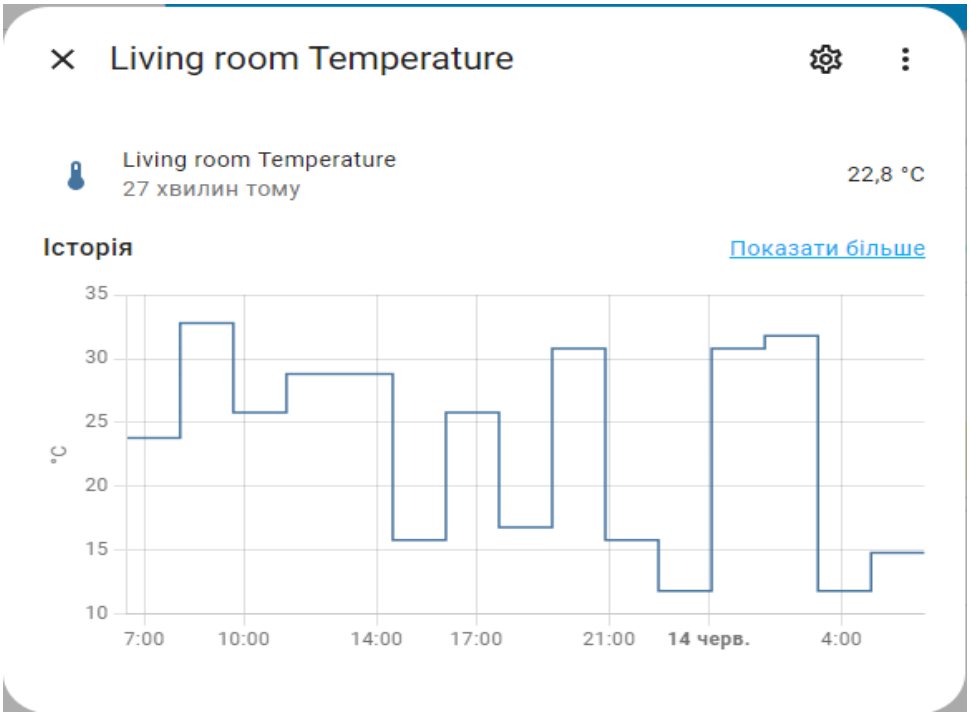

Рисунок 3.9 – Історія зміни температури

Наступна група віджетів має заголовок “Енергія”, вона дозволяє перевіряти і налаштувати інтенсивність вуглекислого газу, обсяг останньої зарядки машини, електроенергію в будинку, вольтаж в будинку та відсоток палива (див. рисунок 3.10).

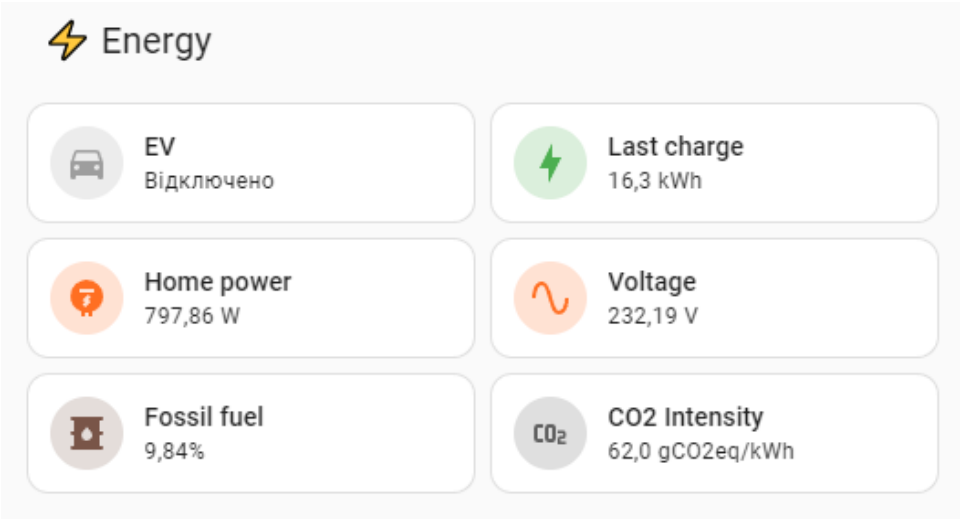

Рисунок 3.10 – Група віджетів “Енергія”

За допомогою ОС Home Assistant можна регулювати температуру вдома, а також отримати інформацію про погоду на вулиці. Налаштування температури вдома дає обрати режим, де є обігрів, авто та вимкнено і один з великої кількості готових пресетів (комфорт, еко, вдома, не вдома). За умови наявності столу, у якого регулюється висота, за допомогою Home Assistant можна переглянути статистику по висоті столу упродовж дня.

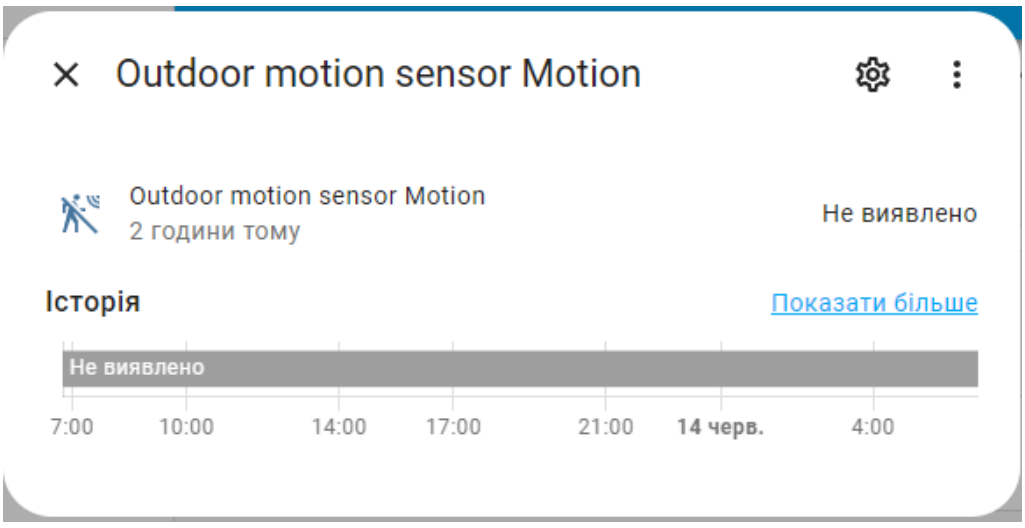

Рисунок 3.11 – Інформація з датчику руху

Передостання група віджетів дозволяє переглянути інформацію про рух за межами будинку за допомогою сенсору (див. рисунок 3.11), а також освітленість за допомогою датчику.

Вкладка “Енергія” дозволяє детальніше оглянути інформацію про енергію в будинку. Для зручності можна обрати будь-який день або період. До прикладу, розглянемо звіт за квітень 2024 року. За допомогою віджетів можна відслідкувати використання та розподіл енергії, де система у кВт вкаже скільки саме були витрачено (див. рисунок 3.12).

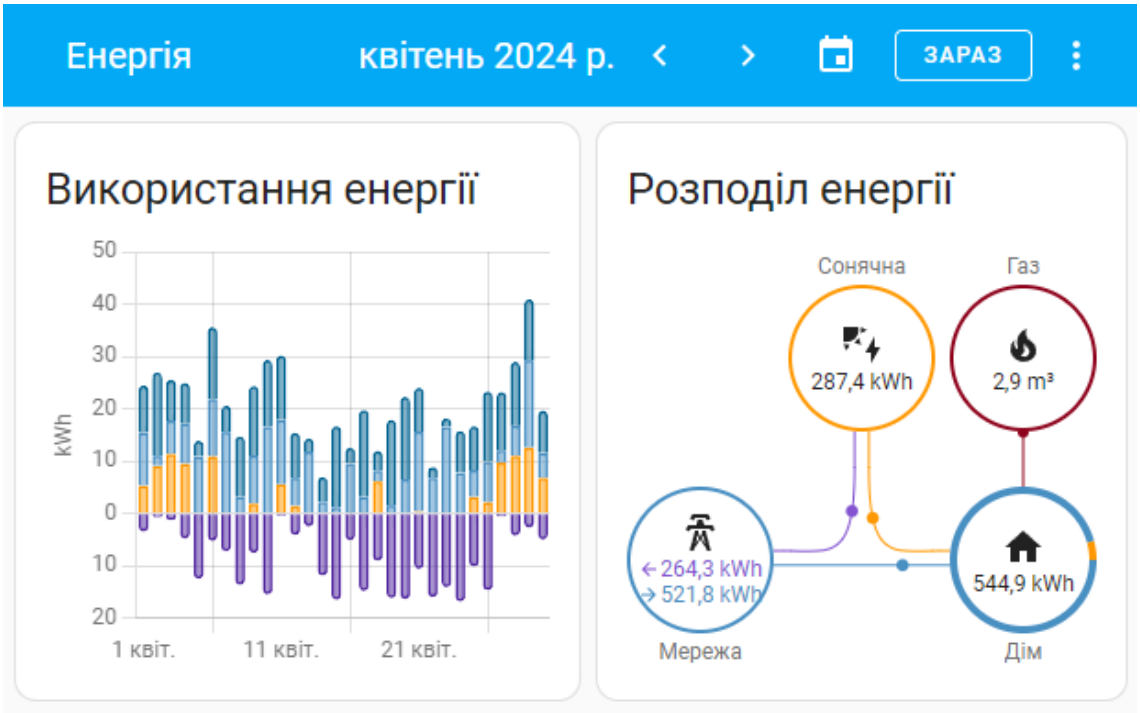

Рисунок 3.12 – Використання та розподіл енергії

За умови, що користувач видобуває сонячну енергію, ОС Home Assistant дозволяє користувачу скільки кВт було вироблено та споживано, а також самодостатність системи у відсотках.

За допомогою системи, можна переглянути об’єм споживаного газу, а також джерела енергії, які використовує Home Assistant. Однією з важливих функцій є можливість перегляду пристроїв, що використовували цю енергію та обсяг у кВт детально та загально.

Вкладка з сповіщеннями показує події, які стаються в межах розумного будинку, до прикладу, якщо датчик зловив якийсь рух на вулиці (див. рисунок 3.13).

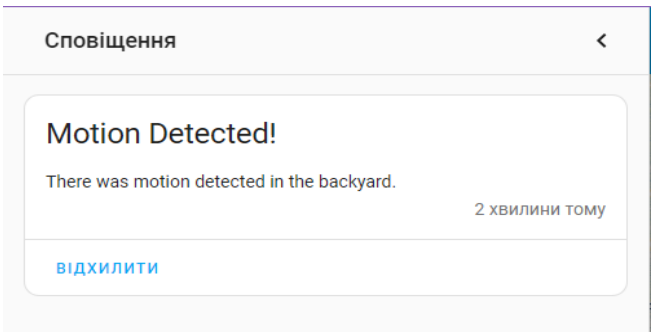

Рисунок 3.13 - Сповіщення

У останній вкладці можна повністю налаштувати систему під свої потреби. З базового – обрати мову, дату та час. Також налаштувати тему браузеру, бічну панель, сповіщення та гарячі клавіші. ОС Home Assistant дозволяє налаштувати та видалити токени оновлення і довгострокові токени доступу (див. рисунок 3.14).

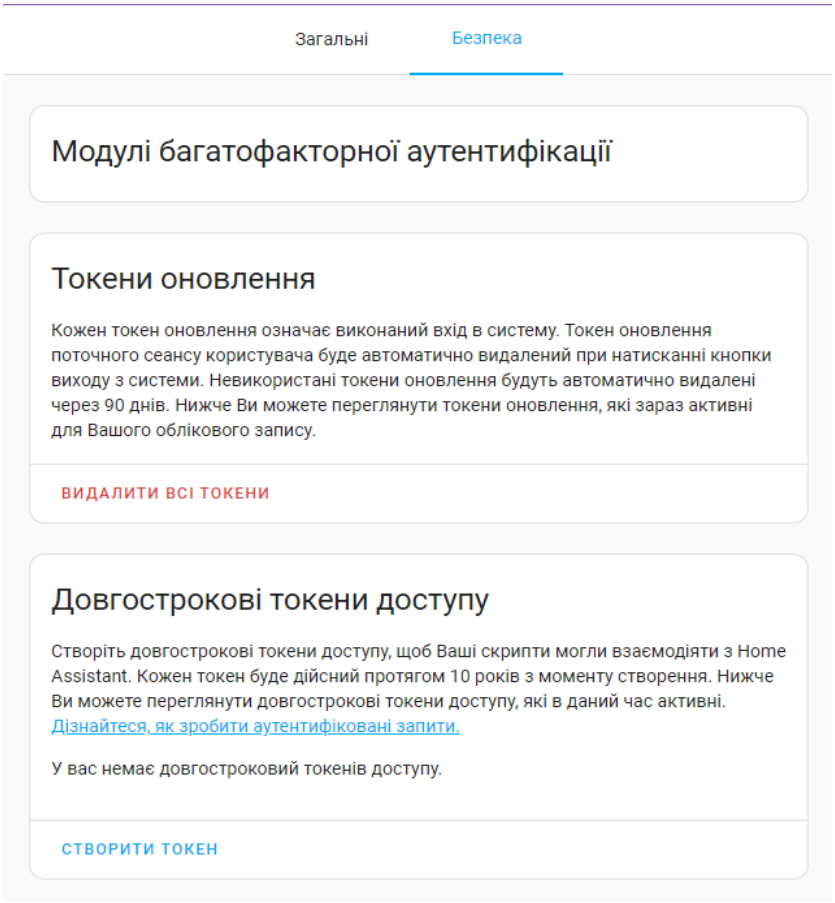

Рисунок 3.14 – Безпека у Home Assistant

Двофакторна аутентифікація додає додатковий рівень безпеки до Home Assistant, вимагаючи ввести код з вашого телефону або іншого пристрою окрім пароля. Хорошим варіантом є брандмауер, що може допомогти захистити систему від несанкціонованого доступу з Інтернету. Також сегментація мережі може допомогти ізолювати Home Assistant від інших пристроїв у мережі, що може ускладнити атаку.

3.2 Захист від XSS-атак

Міжсайтовий скриптинг (XSS) досить поширена вразливість у веб-додатків. Її суть досить проста, зловмисник впроваджує на сторінку код JavaScript, який не було передбачено розробниками. Цей код буде щоразу виконуватись, коли звичайні користувачі будуть заходити на сторінку програми, куди цей код було додано. І в результаті зловмиснику або вдасться отримати дані для авторизації та отримати акаунт або користувача буде направлено на сторінку-клон. Ця веб-сторінка може виглядати ідентично, але належати вона буде зловмиснику. Якщо користувач не помітить різницю, то може ввести важливі дані, які одразу отримає зловмисник.

Home Assistant реалізує кілька механізмів захисту від XSS-атак. Одним з основних методів є очищення HTML-коду за допомогою бібліотеки “bleach”. За допомогою цієї бібліотеки можна видалити небезпечні теги та атрибути, залишивши лише ті, що визначені як безпечні. Функції “filter_xss”, які використовують “bleach”, очищають вхідні дані, видаляючи потенційно шкідливі елементи. Крім того, Home Assistant забезпечує додатковий рівень захисту, дозволяючи реєструвати фільтри шаблонів, щоб очищення HTML-коду автоматично застосовувалося до всіх відповідних шаблонів.

Впровадження таких механізмів безпеки має вирішальне значення для захисту користувачів від XSS-атак. Очищення і перевірка даних, що вводяться користувачем, а також обмеження дозволених тегів і атрибутів можуть значно знизити ризики, пов'язані з виконанням шкідливого коду. Це допомагає зберегти конфіденційність і цілісність даних і забезпечити безперебійну роботу системи.

Оскільки Home Assistant має веб-інтерфейс, то ця вразливість дуже актуальна для системи. Для того, щоб захистити Home Assistant від XSS-атак, був написаний скрипт, що вказаний у лістингу 3.1, який допоможе очистити HTML-дані, що надходять від користувачів.

Лістинг 3.1 – Скрипт для захисту від XSS-атак

```
import bleach
from homeassistant.components import frontend
def filter_xss(html):
    return bleach.clean(html, tags=frontend.ALLOWED_TAGS, attributes=frontend.ALLOWED_ATTRIBUTES)
frontend.register_template_filter('filter_xss', filter_xss)
```

Скрипт використовує бібліотеку bleach для очищення HTML-даних, надісланих користувачем, перед відображенням їх у веб-інтерфейсі Home Assistant. Першим рядком імпортується сама бібліотека, вона дозволяє видаляти або екранувати потенційно небезпечні теги та атрибути. Наступним рядком імпортується компонент “frontend”, який використовується для роботи з інтерфейсом Home Assistant і містить налаштування для дозволених тегів і атрибутів. Функція “filter_xss” приймає рядок HTML на вхід і очищає його за допомогою “bleach.clean”, залишаючи тільки дозволені теги та атрибути, визначені в “frontend.ALLOWED_TAGS” і “frontend.ALLOWED_ATTRIBUTES”. Де властивості функції “html” – вхідний рядок, що містить HTML-код, ‘tags=frontend.ALLOWED_TAGS” – вказує, які HTML-теги дозволені і “attributes=frontend.ALLOWED_ATTRIBUTES” – вказує, які атрибути HTML-тегів дозволені. Останній рядок реєструє функцію “filter_xss” як фільтр шаблону в Home Assistant. Тепер, коли в шаблонах Home Assistant використовується фільтр “filter_xss”, HTML буде очищатися за допомогою цієї функції.

Для використання скрипту його потрібно зберегти як “filter_xss.py” в каталозі “custom_components” Home Asisstant. Далі перезапустити операційну

систему і спокійно використовувати готовий фільтр у шаблонах HTML для очищення даних, що надходять від користувачів.

Приклад використання фільтру вказано у лістингу 3.2.

Лістинг 3.2 – Приклад використання фільтру “filter_xss”

```
- type: markdown
  content: >
    {{ some_html_variable | filter_xss }}
```

Тут “some_html_variable” буде проходити через фільтр “filter_xss”, який видалить або екранує потенційно небезпечні теги та атрибути, захищаючи від XSS-атак.

3.3 Висновки до третього розділу

Веб-інтерфейс ОП Home Assistant дуже зручний та інтуїтивно зрозумілий, що робить його доступним як для початківців, так і для досвідчених користувачів. Home Assistant пропонує гнучку та зручну платформу для домашньої автоматизації, але, як і будь-яка система, вона не є бездоганною. XSS-атаки є серйозною загрозою, яка може призвести до крадіжки даних, перехоплення керування та інших небезпечних наслідків.

Таким чином, розглянуті механізми захисту, що інтегровані в Home Assistant, є ефективним засобом протидії XSS-атакам, підвищують безпеку всієї платформи та довіру користувачів. Це важливий аспект розвитку Home Assistant, спрямований на забезпечення безпечної та стабільної роботи систем автоматизації розумного будинку.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Стихійні лиха та їх класифікація

Землетруси, повені, зсуви, сіли, шторми, торнадо, снігові замети та лісові пожежі забрали життя понад 20 мільйонів людей лише за останні 300 років. За даними Організації Об'єднаних Націй, за цей період близько 10 мільярдів жителів нашої планети постраждали від стихійних лих.

Стихійні лиха - це природні явища, які мають надзвичайний характер та призводять до порушення нормальної діяльності населення, загибелі людей, руйнування і знищення матеріальних цінностей.

За причиною виникнення стихійні лиха поділяють на:

– тектонічні (пов'язані з процесами, які відбуваються в надрах земної кори);

– топологічні (пов'язані з процесами, які відбуваються на поверхні землі);

– метеорологічні (пов'язані з процесами, які відбуваються в атмосфері).

Особливості географічного положення України, атмосферні процеси, наявність гірських масивів, підвищень, близькість теплих морів зумовлюють різноманітність кліматичних умов: від надлишкового зволоження в західному Поліссі - до посушливого - в південній Степовій зоні. Виняткові кліматичні умови на Південному березі Криму, в горах Українських Карпат та Криму.

Внаслідок взаємодії всіх цих факторів виникають небезпечні стихійні явища. В окремих випадках вони мають катастрофічний характер.

Стихійні явища часто виникають в комплексі, що значно посилює їх негативний вплив. Небезпечні природні явища, переважно, визначаються трьома основними групами процесів - ендогенні, екзогенні та гідрометеорологічні.

Стихійні лиха, що мають місце на території України, можна поділити на прості, що включають один елемент наприклад, сильний вітер, зсув або землетрус) та складні, що включають декілька процесів однієї групи або кількох груп, наприклад, негативних атмосферних та геодинамічних екзогенних

процесів, ендогенних, екзогенних та гідрометеорологічних процесів у поєднанні з техногенними [30].

Землетруси - коливання земної кори, що виникають внаслідок вибухів вглибині землі, розламів шарів земної кори, активної вулканічної діяльності. Область підземного удару викликає пружні коливання (сейсмічні хвилі), що поширюються по землі у всіх напрямках. Область землі, з якої виходять хвилі землетрусу, називають центром, а розташовану на поверхні землі ділянку - епіцентром землетрусу.

Інтенсивність землетрусу вимірюється в балах за шкалою Ріхтера, а у останні роки у нашій країні та у ряді європейських держав використовують 12-бальну міжнародну шкалу МЗК-64 [32]. Інтенсивність землетрусу зменшується до периферії зони катастрофи. Осередки землетрусів знаходяться на глибині 30-60 км, а інколи на глибині до 700 км. В залежності від причин і місця виникнення землетруси поділяються на тектонічні, вулканічні, обвальні і моретруси.

Землетруси захоплюють великі території і характеризуються:

- руйнуванням будівель і споруд, під уламки яких потрапляють люди, виникненням масових пожеж і виробничих аварій;
- затопленням населених пунктів і цілих районів;
- отруєнням газами при вулканічних виверженнях;
- ураженням людей і руйнуванням будівель уламками вулканічних гірських порід;
- ураженням людей і виникненням осередків пожеж у населених пунктах від вулканічної лави;
- провалом населених пунктів при обвальних землетрусах;
- руйнуванням і змиванням населених пунктів хвилями цунамі;
- негативною психологічною дією.

Серед всіх стихійних лих за даними ЮНЕСКО землетруси займають перше місце в світі за заподіяною економічною шкодою і кількістю загиблих.

Вулканізм - це сукупність явищ, зумовлених проникненням магми з глибини землі на її поверхню. Процеси грязьового вулканізму локалізовані у

південній частині території України. Вони спостерігаються на Керченському півострові та прилеглій акваторії Азовського моря [33].

Якщо оцінювати площу України з точки зору негативних екзогенних природних процесів, можна виділити площі з різним ступенем ризику виникнення природного (або стихійного) лиха.

Широкий розвиток мають різні види екзогенних геологічних процесів природного та техногенного походження.

Сель (паводок) - раптово сформований, внаслідок різкого підйому води в руслах гірських річок, грязьовий потік. Причинами виникнення селевих потоків майже завжди бувають сильні зливи, інтенсивне танення снігу та льоду, прорив гребель водойм, а також землетруси та виверження вулканів. Сель характеризується значною руйнівною силою ґрунту та каміння, що насувається, виникає раптово, рухається зі швидкістю понад 10 м/с [34].

Виникненню їх сприяють і антропогенні фактори: вирубка лісів і деградація ґрунтів на гірських схилах, вибухи гірських порід при прокладанні доріг, роботи у кар'єрах, неправильна організація обвалів та підвищень, загазованість повітря, що згубно діє на ґрунтово-рослинний покрив.

Протягом останнього десятиріччя в Україні зафіксовано близько 240 випадків виникнення катастрофічних природних явищ метеорологічного походження зі значними матеріальними збитками.

Небезпечні метеорологічні явища, що мають місце в Україні:

- сильні зливи (Карпатські та Кримські гори);
- град (на всій території України);
- сильна спека (степова зона);
- суховії, посухи (степова та східна лісостепова зони);
- урагани, шквали, смерчі (більша частина території);
- пилові бурі (південний схід степової зони);
- сильні тумани (південний схід степової зони);
- сильні заметілі (південний схід степової зони);
- снігові заноси (Карпати);
- значні ожеледі (степова зона);

– сильний мороз (північ Полісся та схід лісостепової зони).

Крім того, вздовж узбережжя та в акваторії Чорного і Азовського моря мають місце шторми, ураганні вітри, смерчі, зливи, обмерзання споруд та суден, сильні тумани, заметілі, ожеледі [33].

Щорічно в суху, жарку погоду небезпека від лісових та торф'яних пожеж різко зростає. Лісові пожежі виникають головним чином з вини людини та внаслідок дії деяких природних чинників. Причиною пожеж буває виробнича діяльність людини (спалювання відходів на прилеглих до лісу територіях) та її необережність (вогнища, недопалки, сірники).

Вогонь може швидко розростися і, підхоплений вітром, стати вогненним валом, що знищує на своєму шляху все живе і перетворює ліси в нежиттєздатні пустелі. При цьому виникає, велика загроза населеним пунктам, життю людей, домашнім тваринам, матеріальним цінностям. Найбільш небезпечними бувають жаркі та сухі літні дні з відносною вологістю повітря 30-40%.

Залежно від характеру горіння, швидкості розповсюдження вогню та розмірів пошкодження лісу розрізняють чотири категорії лісових пожеж:

– низові (або низинні);
– верхові (або повальні);
– підземні (торф'яні або ґрунтові);
– пожежі дуплистих дерев.

Найбільш поширеними є лісові та торф'яні пожежі, бо ліси і торфовища займають більше 10 млн. га території України. 31% лісів розташовано в північному регіоні, 17% - в східному, 10% - в південному, 8% - в південно-західному і 32% - в західному регіоні. Лісовий фонд України майже на 50% складається з хвойних лісів, з яких 60% займають молодняки [34].

4.2 Інженерно-технічні рішення з охорони праці

До категорії заходів попередження нещасних випадків слід віднести такі:

– при експлуатації пристроїв, обчислювальних машин, комплексів, систем і мереж;

– у технологічних процесах та при експлуатації апаратури збору й відображення інформації.

Для захисту людей від ураження електричним струмом слід передбачати блокувальні пристрої, електричний розподіл мереж, занулення, подвійну ізоляцію, захисне вимикання [35]. Вибір захисних пристроїв потрібно обґрунтувати з посиланням на нормативні документи та навести його (їх) схему. При розробці основних вимог щодо електропроводки виробничого приміщення треба за встановленою потужністю споживачів вибрати тип і кількість силових кабелів, встановити місце розташування поживлюючого щита, вибрати пускорегулюючу та захисну апаратуру. Особливу увагу необхідно приділити забезпеченню швидкого вимикання пристроїв обчислювальної машини в разі аварії або нещасного випад-ку. Якщо необхідно, слід вказати причини виникнення статичної електрики у виробничому приміщенні, величину його потенціалу, розробити заходи щодо попередження утворення статичних зарядів, їх нейтралізації та зняття [36].

Необхідно описати умови експлуатації апаратури збору інформації, передбачити захист цієї апаратури від можливого агресивного середовища, пилу, вологи, променевого тепла тощо, вибрати засіб живлення та електричного захисту цієї апаратури. Якщо домінуючим є небезпечний виробничий чинник, то необхідно провести розрахунок захисних пристроїв від нього. Для приміщень, в яких використовується ПОЕМ, лабораторій та інших виробничих приміщень визначають основні джерела виділення надмірного тепла та сумарне тепловиділення. Вибирають засіб вилучення надмірного тепла, обґрунтовують необхідність кондиціонування повітря, здійснюють вибір необхідного обладнання (наводять повну технічну характеристику).

Необхідно також вибрати схему циркуляції повітря (через підпідлоговий простір, або простір над підвісною стелею). При проєктуванні штучного освітлення в приміщеннях необхідно керуватися вимогами ДНАОП 0.00-1.31-99, СНиП II-4-79. При цьому при встановленні нормативної освітленості (на робочих місцях, в проходах, аварійної), вибирають систему освітлення, тип освітлювачів і ламп, визначають їхню кількість та розміщення. У випадку

перевищення рівнів звукового тиску в приміщенні, порівняно з нормативним, передбачають заходи з поліпшення шумового режиму: екранування принтерів, облицювання стелі та стін звукопоглинаючим матеріалом (навести технічну характеристику). Якщо домінуючим є шкідливий виробничий чинник, то необхідно провести розрахунок захисних пристроїв від нього Організація та конструкція робочого місця користувача ЕОМ має забезпечувати відповідність всіх елементів робочого місця і їхнього розташування ергономічними вимогам ГОСТ 12.2.032.-78 та ДНАОП 0.00-1.31-99. Тому необхідно навести цю відповідність та схему розміщення робочих місць у приміщенні.

Необхідно мати на увазі, що режим праці і відпочинку працюючих з ЕОМ визначається у залежності від виконуваної категорії роботи. Тому необхідно визначити належність виконуваних робіт до однієї з трьох груп трудової діяльності: група А – діяльність, яка характеризується виконанням одноманітних, ритмічних, легких у виконанні операцій, що не вимагають значної розумової напруги; група Б – діяльність, пов'язана зі здійсненням повторюваних логічних операцій; група В – творчі види діяльності, що вимагають прийняття у процесі роботи рішень за відсутністю заздалегідь відомого алгоритму.

На підставі цього встановити раціональний режим праці та відпочинку, додаткові перерви. Також необхідно визначити рівень навантаження за робочу зміну: кількість знаків за робочу зміну (у тисячах) або тривалість роботи за зміну (годин). Вимоги до організації робочого місця та режиму роботи мають приводити психофізіологічні НШВЧ до норм.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проаналізовано механізми безпеки та реалізовано архітектуру захисту операційної системи Home Assistant від XXS-атак.

В ході виконання першого розділу кваліфікаційної роботи було проаналізовано галузь застосування та механізми безпеки ОС Home Assistant. Були розглянуті механізми автентифікації та авторизації, шифрування комунікації, захист від атак, аудит та журналювання.

У другому розділі кваліфікаційної роботи було розглянуто протоколи для шифрування, серед яких SSL/TLS, HTTPS, MQTTs та SSH. А також захисти від атак переповнення буфера, SQL-ін’єкцій і CSRF-атак. Було проаналізовано принципи найменшого доступу та розділення обов’язків.

Третій розділ кваліфікаційної роботи присвячений виконанню поставленого завдання. Для його реалізації було налаштовано внутрішню архітектуру безпеки операційної системи Home Assistant, зокрема реалізація використання операційної системи на віртуальній машині. Для безпосереднього втілення задачі був написаний скрипт “filter_xss”, який дозволяє очищувати HTML-дані, надіслані користувачем, перед відображенням їх у веб-інтерфейсі Home Assistant, для захисту ОС від XSS-атак.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Beshley, M. Development of a cyber-physical system for automation and control of the Internet of things using the Home Assistant platform / M. Beshley, Y. Shkoropad, H. Beshley // Information and communication technologies, electronic engineering. – 2024. – Vol. 4, no. 1. – P. 20–30.

2. Kolodiichuk, L. Using the Home Assistant Digital Platform to Control the Electrical Installation / L. Kolodiichuk // Energy & Automation. – 2023. – №1.

3. Introducing Hass.io. Home Assistant. URL: https://www.home-assistant.io/blog/2017/07/25/introducing-hassio/

4. Kharchenko, V. Dependable IoT for Human and Industry: Modeling, Architecting, Implementation / V. Kharchenko, A.L. Kor, A. Rucinski, (Eds.) // CRC Press. – 2022.

5. Skarga-Bandurova, I. Iot For Public Transport Information Service Delivering / I. Skarga-Bandurova, M. Derkach // Internet of Things for Industry and Human Applications. Volume 3. Assessment and Implementation. Intelligent Transportation Systems and IoT. Section 41. / Ed. V. S. Kharchenko. – Ministry of Education and Science of Ukraine, National Aerospace University KhAI. – 2019. – P. 373-401.

6. Palamar, A. Remote Air Pollution Monitoring System Based on Internet of Things / A. Palamar, M.P. Karpinski, M. Palamar, H. Osukhivska, M. Mytnyk // In 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP). – 2022. – P. 194-204.

7. IKEA Tradfri documentation URL: https://www.ikea.com/gb/en/customer-service/product-support/app-gateway/getting-started-and-how-to-use-pubce3e6297

8. Home Assistant Community URL: https://community.home-assistant.io/t/bathroom-humidity-exhaust-fan/509992

9. Mishko, O. Security of remote IoT system management by integrating firewall configuration into tunneled traffic / O. Mishko, D. Matiuk, M. Derkach // Scientific Journal of TNTU. — Tern.: TNTU, 2024.

10. Посібник yii framework URL: https://yiiframework.com.ua/uk/doc/guide/topics.auth/

11. Farooq, M. Stout Implementation of Firewall and Network Segmentation for Securing IoT Devices / M. Farooq, R. Khan, M.H. Khan // Indian Journal of Science and Technology. – 2023. – №16(33). – P. 2609-2621.

12. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). ЗАСТОСУВАННЯ ГІПЕРВІЗОРІВ ПЕРШОГО ТИПУ ДЛЯ СТВОРЕННЯ ЗАХИЩЕНОЇ ІТ-ІНФРАСТРУКТУРИ. Матеріали конференцій МЦНД, (24.05. 2024; Запоріжжя, Україна), 145-146. https://doi.org/10.62731/mcnd-24.05.2024.001

13. Karnaukhov, A., Tymoshchuk, V., Orlovska, A., & Tymoshchuk, D. (2024). USE OF AUTHENTICATED AES-GCM ENCRYPTION IN VPN. Матеріали конференцій МЦНД, (14.06. 2024; Суми Україна), 191-193. https://doi.org/10.62731/mcnd-14.06.2024.004

14. Офіційний сайт Home Assistant. Захист URL: https://www.home-assistant.io/docs/configuration/securing/

15. Офіційна документація Home Assistant з питань журналювання URL: https://www.home-assistant.io/integrations/logger/

16. DEV Community URL: https://dev.to/techschoolguru/a-complete-overview-of-ssl-tls-and-its-cryptographic-system-36pd

17. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). СИСТЕМА ЗМЕНШЕННЯ ВПЛИВУ DOS-АТАК НА ОСНОВІ MIKROTIK. Матеріали конференцій МЦНД, (17.05. 2024; Ужгород, Україна), 198-200. https://doi.org/10.62731/mcnd-17.05.2024.008

18. Buffer Overflow URL: https://www.fortinet.com/resources/cyberglossary/buffer-overflow

19. Yesin, V., Karpinski, M., Yesina, M., Vilihura, V., Kozak, R., Shevchuk, R. (2023). Technique for Searching Data in a Cryptographically Protected SQL Database. Applied Sciences, 13(20), art. no. 11525, 1-21. doi: 10.3390/app132011525.

20. Бекер, І., Тимощук, В., Маслянка, Т., & Тимощук, Д. (2023). МЕТОДИКА ЗАХИСТУ ВІД ПОВІЛЬНИХ ТА ШВИДКИХ BRUTE-FORCE АТАК НА IMAP СЕРВЕР. Матеріали конференцій МНЛ, (17 листопада 2023 р., м. Львів), 275-276.

21. Cross Site Request Forgery URL: https://owasp.org/www-community/attacks/csrf

22. Ванца, В., Тимощук, В., Стебельський, М., & Тимощук, Д. (2023). МЕТОДИ МІНІМІЗАЦІЇ ВПЛИВУ SLOWLORIS АТАК НА ВЕБСЕРВЕР. Матеріали конференцій МЦНД, (03.11. 2023; Суми, Україна), 119-120.

23. What is Access Control List (ACL). Sangfor Technologies. URL: https://www.sangfor.com/glossary/cybersecurity/what-is-access-control-list-acl

24. Іваночко, Н., Тимощук, В., Букатка, С., & Тимощук, Д. (2023). РОЗРОБКА ТА ВПРОВАДЖЕННЯ ЗАХОДІВ ЗАХИСТУ ВІД UDP FLOOD АТАК НА DNS СЕРВЕР. Матеріали конференцій МНЛ, (3 листопада 2023 р., м. Вінниця), 177-178.

25. Демчук, В., Тимощук, В., & Тимощук, Д. (2023). ЗАСОБИ МІНІМІЗАЦІЇ ВПЛИВУ SYN FLOOD АТАК. Collection of scientific papers «SCIENTIA», (November 24, 2023; Kraków, Poland), 130-130.

26. SSL Security URL: https://avinetworks.com/glossary/ssl-security/

27. What`s the Difference Between HTTP and HTTPS? URL: https://aws.amazon.com/compare/the-difference-between-https-and-http/?nc1=h_ls

28. MQTTS: How to use MQTT with TLS? URL: https://openest.io/en/services/mqtts-how-to-use-mqtt-with-tls/

29. What is SSH and how it works? URL: https://introserv.com/ua/blog/scho-take-protokol-ssh-i-yak-vin-praczyue/

30. Дмітрієв Ю. Д. Охорона навколишнього середовища. – К.: Вища школа, 1997. – 189с.

31. Tymoshchuk, V., Dolinskyi, A., & Tymoshchuk, D. (2024). MESSENGER BOTS IN SMART HOMES: COGNITIVE AGENTS AT THE FOREFRONT OF THE INTEGRATION OF CYBER-PHYSICAL SYSTEMS AND THE INTERNET OF THINGS. Матеріали конференцій МЦНД, (07.06.2024; Луцьк, Україна), 266–267. https://doi.org/10.62731/mcnd-07.06.2024.004

32. ДСТУ Б В.1.1-28:2010

33. Єлін Ю. Я., Зерова М. Я. Україна: екологічний аспект. – К.: Просвіта, 1995. – 217с.

34. Ємченко О. П. Безпека життєдіяльності. – Харків: Астра-Пресс, 1996. – 98с.

35. ГОСТ 12.1. 019-79

36. ГОСТ 12.4.124-83

Додаток А

Бібліотека bleach

```
#!/usr/bin/env python

import os
import re

from setuptools import setup, find_packages


def get_long_desc():
    with open("README.rst", encoding="utf-8") as fp:
        desc = fp.read()
    desc += "\n\n"
    with open("CHANGES", encoding="utf-8") as fp:
        desc += fp.read()
    return desc


def get_version():
    fn = os.path.join("bleach", "__init__.py")
    vsre = r"""^__version__ = ['"]([^'"]*)['"]"""
    with open(fn, encoding="utf-8") as fp:
        version_file = fp.read()
    return re.search(vsre, version_file, re.M).group(1)


INSTALL_REQUIRES = [
    # html5lib requirements
    "six>=1.9.0",
    "webencodings",
]


EXTRAS_REQUIRE = {
    "css": [
        "tinycss2>=1.1.0,<1.3",
    ],
}


setup(
    name="bleach",
    version=get_version(),
    description="An easy safelist-based HTML-sanitizing tool.",
    long_description=get_long_desc(),
    long_description_content_type="text/x-rst",
    maintainer="Will Kahn-Greene",
    maintainer_email="willkg@mozilla.com",
    url="https://github.com/mozilla/bleach",
```

```
    license="Apache Software License",
    packages=find_packages(),
    include_package_data=True,
    package_data={"": ["README.rst"]},
    zip_safe=False,
    python_requires=">=3.8",
    install_requires=INSTALL_REQUIRES,
    extras_require=EXTRAS_REQUIRE,
    classifiers=[
        "Development Status :: 5 - Production/Stable",
        "Environment :: Web Environment",
        "Intended Audience :: Developers",
        "License :: OSI Approved :: Apache Software License",
        "Operating System :: OS Independent",
        "Programming Language :: Python",
        "Programming Language :: Python :: 3 :: Only",
        "Programming Language :: Python :: 3",
        "Programming Language :: Python :: 3.8",
        "Programming Language :: Python :: 3.9",
        "Programming Language :: Python :: 3.10",
        "Programming Language :: Python :: 3.11",
        "Programming Language :: Python :: 3.12",
        "Programming Language :: Python :: Implementation :: CPython",
        "Programming Language :: Python :: Implementation :: PyPy",
        "Topic :: Software Development :: Libraries :: Python Modules",
    ],
)
```