Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра комп'ютерних систем та технологій

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Розробка портативного хот-спота з функціями мережного екрану на базі raspberry pi»

«Development of portable hotspot with firewall functions based on raspberry pi»

Виконав: здобувач заочної форми навчання
спеціальності 123 – Комп'ютерна інженерія
Освітня програма «Комп'ютерна інженерія»

Смирнов Владислав Геннадійович
Керівник __ст. вик. Берков Юрій Миколайович__
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)
Рецензент ___д.т.н. проф._ГУНЧЕНКО_Ю.О.___
(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
№ ___ від _____ р.

Завідувач кафедри

_____ Юрій ГУНЧЕНКО
(підпис) (прізвище, ім'я)

Захищено на засіданні ЕК №
протокол № ____ від _____ р.
Оцінка_____/___ ___/_____
(за національною шкалою, шкалою ECTS, бали)
Голова ЕК

_____ Алла КОБОЗЄВА
(підпис) (прізвище, ім'я)

Одеса – 2023

АНОТАЦІЯ

Дипломна робота присвячена «Розробка портативного хот-спота з функціями мережного екрану на базі raspberry pi». Метою дослідження є створення зручного та функціонального пристрою, який поєднує мобільність та можливості мережевого екрану.

У роботі проведено аналіз сучасного стану справ у сфері портативних хот-спотів та функцій мережного екрану. Пояснено актуальність теми, висвітлені проблеми та виклики, які потребують уваги та розв'язання. Також приведені потенційні переваги та користь від розробки портативного хот-споту на базі Raspberry Pi.

У роботі описано задачі, які були вирішені для досягнення мети дослідження, зокрема вибір апаратного забезпечення, розробка програмного забезпечення, налаштування мережевих функцій та інтерфейсу користувача. Також проведено тестування та оцінку пристрою з метою перевірки його ефективності та надійності.

Висновки дослідження свідчать про успішну «Розробка портативного хот-спота з функціями мережного екрану на базі raspberry pi». Цей пристрій дозволяє забезпечити безпечний та контрольований доступ до Інтернету в будь-якому місці, забезпечуючи захист від шкідливого контенту. Він має потенціал для застосування в різних сферах, включаючи мобільний Інтернет, подорожі та роботу віддалено.

ABSTRACT

The thesis is dedicated to the "Development of a portable hotspot with network firewall features based on Raspberry Pi." The aim of the research is to create a convenient and functional device that combines mobility and network firewall capabilities.

The study includes an analysis of the current state of portable hotspots and network firewall functions. The relevance of the topic is explained, and the problems and challenges requiring attention and resolution are highlighted. Potential advantages and benefits of developing a portable hotspot based on Raspberry Pi are also presented.

The thesis describes the tasks that were accomplished to achieve the research goal, including hardware selection, software development, configuration of network functions and user interface. Testing and evaluation of the device were conducted to assess its effectiveness and reliability.

The research findings indicate the successful development of a portable hotspot with network firewall features based on Raspberry Pi. This device enables secure and controlled Internet access anywhere while providing protection against harmful content. It has the potential for applications in various domains, including mobile Internet, travel, and remote work.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

IDS (англ. Intrusion Detection System, IDS) - Система виявлення атак.

IPS (англ. Intrusion Prevention System, IPS) - Система запобігання вторгненням.

GPIO (англ. General-purpose input/output, GPIO) -Інтерфейс введення/виведення загального призначення

USB (англ. Universal Serial Bus) - універсальна послідовна шина.

МБ – мегабайт (220 байт)

HTML (англ. HyperText Markup Language) — мова розмітки гіпертексту

CSS (англ. Cascading Style Sheets) — Каскадні таблиці стилів

ВСТУП

У сучасному світі, де доступ до Інтернету став необхідністю, бездротові мережі стали невід'ємною частиною нашого повсякденного життя. Однак, є ситуації, коли ми опиняємось далеко від стаціонарних мереж або не маємо доступу до мобільної мережі. У таких випадках розробка портативного хот-спота, який надає зручний доступ до Інтернету та функції мережного екрану, стає важливим завданням.

Міні-комп'ютер Raspberry Pi є потужним та універсальним інструментом для реалізації цієї ідеї. Він має компактний розмір, доступну ціну та відкрите програмне забезпечення, що дозволяє налаштувати його під свої потреби.

Розробка портативного хот-спота на базі Raspberry Pi дозволяє створити переносний пристрій, який надає надійний та зручний доступ до Інтернету в будь-який час і в будь-якому місці. Окрім цього, він може виконувати функції мережного екрану, дозволяючи контролювати та обмежувати доступ до Інтернету для забезпечення безпеки та конфіденційності.

Цей проект передбачає налаштування бездротового з'єднання, створення мережі та налаштування функцій мережного екрану. Результатом роботи буде функціональний та зручний портативний хот-спот, який забезпечуватиме стабільний доступ до Інтернету та контроль над підключеними пристроями.

Розробка портативного хот-спота з функціями мережного екрану на базі Raspberry Pi має великий потенціал для поліпшення зв'язку та розширення можливостей доступу до Інтернету. Вона відкриває нові перспективи для розвитку технологій і дозволяє нам бути підключеними в будь-який час і в будь-якому місці. Завдяки цьому проекту ми зможемо насолоджуватись швидким та зручним доступом до Інтернету, забезпечуючи безпеку та надійність з'єднання.

Мета даного дослідження полягає в розробці портативного хот-споту

на базі Raspberry Pi з функціями мережного екрану. Цей пристрій має на меті забезпечити користувачам зручний та безпечний доступ до Інтернету у будь-якому місці.

Для досягнення поставленої мети передбачається виконання наступних завдань:

1) Вибір апаратного забезпечення: Проведення дослідження та аналіз різних моделей Raspberry Pi для вибору найбільш підходящої для розробки портативного хот-споту. Оцінка характеристик процесора, пам'яті, безпроводових модулів та інших компонентів, які впливають на продуктивність та функціональні можливості пристрою.
2) Розробка програмного забезпечення: Програмування та налаштування необхідного програмного забезпечення для забезпечення роботи хот-споту та мережного екрану. Це включає налаштування мережевих протоколів, реалізацію безпеки мережі, управління трафіком та інші функції, що забезпечують ефективну та безпечну роботу пристрою.
3) Налаштування мережевих функцій та інтерфейсу користувача: Забезпечення можливості підключення до хот-споту через різні пристрої, налаштування параметрів безпеки та контролю доступу. Розробка інтуїтивно зрозумілого та зручного інтерфейсу користувача, який дозволяє налаштовувати та керувати пристроєм.
4) Тестування та оцінка пристрою: Проведення випробувань та тестувань розробленого хот-споту з метою перевірки його функціональності, надійності та продуктивності. Зіставлення результатів тестування з поставленими вимогами та стандартами для оцінки ефективності пристрою.

Детальне виконання цих завдань дозволить реалізувати мету дослідження - розробку портативного хот-споту з функціями мережного екрану на базі Raspberry Pi.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ АПАРАТНОГО ЗАБЕСПЕЧЕННЯ

Raspberry Pi - це одноплатний комп'ютер, розроблений з метою сприяти навчанню програмуванню та експериментам з комп'ютерною технологією. Він базується на системі-на-чипі Broadcom SoC (System-on-a-Chip) і має вбудовану пам'ять, процесор, відеокарту, аудіокарту та інші периферійні пристрої.

1.1 Характеристики Raspberry Pi

Основні характеристики Raspberry Pi залежать від конкретної моделі, оскільки існує кілька поколінь та версій цього пристрою. Найпопулярнішою моделлю є Raspberry Pi 4, який має такі технічні характеристики:

- Процесор: 1,5 ГГц чотирьохядерний 64-бітовий ARM Cortex-A72;
- Пам'ять: 2, 4 або 8 ГБ оперативної пам'яті LPDDR4-3200;
- Графічний процесор: Вбудований графічний процесор VideoCore VI з підтримкою 4K відео;
- Порти: 2 порти USB 3.0, 2 порти USB 2.0, 2 порти micro-HDMI з підтримкою 4K відео, роз'єм Ethernet, роз'єм для карт пам'яті microSD, аудіовихід, роз'єм GPIO та інші. На рисунку нижче показано плату Raspberry PI (рис. 1.1).

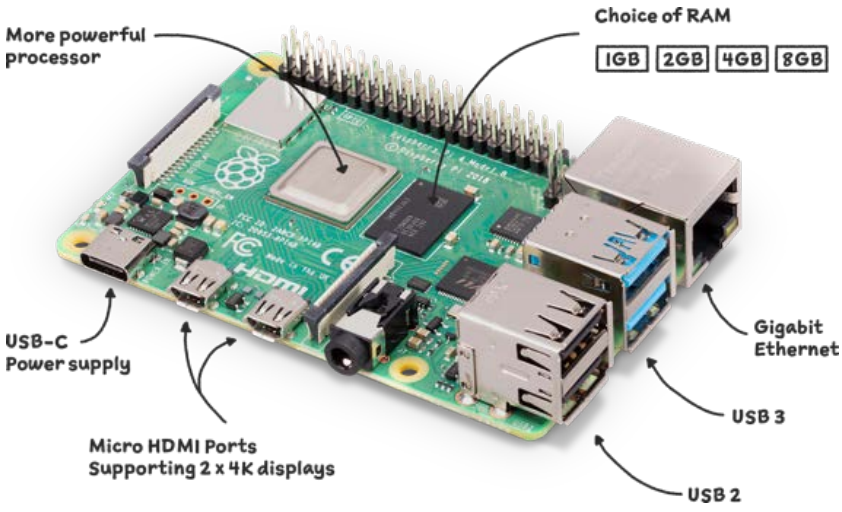


Рисунок 1.1 – “Raspberry Pi”

Raspberry Pi працює на базі операційної системи Raspbian, яка заснована на Linux. Однак, існують також інші операційні системи, які можна встановити на Raspberry Pi, включаючи Ubuntu, Fedora, Kali Linux та інші.

Основна перевага Raspberry Pi полягає в його доступності та гнучкості. Цей пристрій може бути використаний для різних проектів, таких як домашній медіацентр, інтернет-сервер, система моніторингу, розумний дім, робототехніка та багато іншого. Завдяки великій спільноті розробників і доступності документації, Raspberry Pi став популярним інструментом для навчання, творчості та експериментів з комп'ютерною технологією.

Raspberry Pi має кілька переваг, які виокремлюють його серед інших аналогів:

Вартість: Raspberry Pi пропонує високий рівень функціональності за досить низьку ціну. Це робить його доступним для широкого кола користувачів, включаючи студентів, гобістів та малі підприємства.

Відкрите програмне забезпечення: Raspberry Pi базується на операційній системі Linux та має велику спільноту розробників, що розвиває відкрите програмне забезпечення. Це означає, що користувачі мають доступ до безлічі безкоштовних програм, додатків і ресурсів, що розширюють можливості пристрою.

Гнучкість: Raspberry Pi може бути використаний для різних проектів та завдань. Він може бути використаний як домашній сервер, медіацентр, IoT-платформа, система автоматизації, контролер роботів та багато іншого. Його загальні розміри та низька споживання енергії роблять його зручним для використання в різних середовищах.

1.2 GPIO інтерфейси

Розширюваність: Raspberry Pi має роз'єм GPIO (General Purpose Input/Output), який дозволяє підключати різні датчики, модулі та розширювальні плати. Це дає користувачам можливість створювати свої

власні пристрої та розширювати функціональність Raspberry Pi. На рисунку 1.2 показаний GPIO інтерфейс.

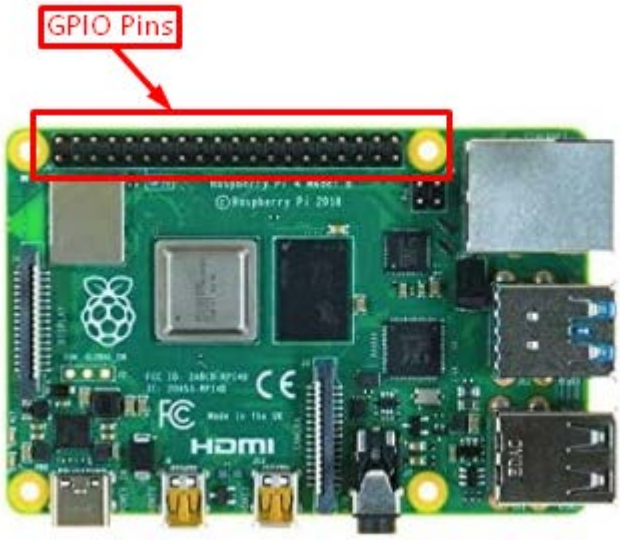


Рисунок 1.2 – GPIO інтерфейс

Навчання та освіта: Raspberry Pi був спеціально розроблений з метою підтримки навчання програмування та комп'ютерної технології. Він широко використовується у школах та навчальних закладах для навчання студентів основам програмування та електроніки.

Активна спільнота: Raspberry Pi має велику та активну спільноту користувачів та розробників. Це означає, що завжди є доступ до підтримки, порад та розв'язань проблем. Можна отримати допомогу та поради від інших користувачів, а також долучитися до проектів спільноти.

Ці переваги роблять Raspberry Pi популярним вибором для широкого кола застосувань, починаючи від освіти та домашніх проектів до промислових та комерційних застосувань[2].

2 РЕАЛІЗАЦІЯ HOT-SPOT НА БАЗІ RASPBERRY PI

В цьому розділі описується процес реалізації портативного хот-споту. Проводиться налаштування бездротового з'єднання та робота з Wi-Fi на Raspberry Pi. Розробляється програмне забезпечення, яке дозволить керувати хот-спотом, включаючи налаштування параметрів мережі, керування доступом та моніторинг стану хот-споту. Також проводиться тестування та налагодження системи з метою впевненості в її правильному функціонуванні.

Для реалізації hot-spot на базі Raspberry Pi використовувались такі мови програмування такі як python, php, javascript, та мова розмітки html та css.

2.1 Аналіз веб-серверів

Було прийняте рішення використовувати ядро веб-серверу Lighttpd замість flask. Також для DHCP сервера був вибраний dnsmasq Lighttpd і Flask - це дві різні технології, які використовуються в різних аспектах веб-розробки. Ось три ключові відмінності між ними:

Функціональність:

Lighttpd: Lighttpd є веб-сервером, який відповідає за обробку запитів, передачу файлів та керування з'єднаннями. Він призначений для обслуговування статичного та динамічного вмісту, включаючи сторінки HTML, CSS, JavaScript та інші ресурси.

Flask: Flask - це мінімалістичний фреймворк для розробки веб-додатків з використанням мови програмування Python. Він надає основну структуру та інструменти для розробки веб-додатків, включаючи маршрутизацію URL, обробку запитів та відповідей, шаблонізацію, обробку форм та інші функціональності.

Рівень абстракції:

Lighttpd: Lighttpd працює на рівні веб-сервера і взаємодіє безпосередньо з клієнтами, оброблюючи HTTP-запити та передаючи відповіді. Він не надає специфічних інструментів для розробки додатків, а

лише забезпечує середовище для їх виконання.

Flask: Flask - це фреймворк, який вище рівня, надаючи інструменти для створення веб-додатків. Він надає абстракцію над рівнем веб-сервера та надає зручний спосіб визначення маршрутів, обробки запитів, роботи з шаблонами та інші функціональності, які полегшують розробку веб-додатків.

Мова програмування:

Lighttpd: Lighttpd не пов'язаний з конкретною мовою програмування. Він може працювати з будь-якими мовами, що підтримують взаємодію через протоколи HTTP або FastCGI/SCGI.

Flask: Flask розроблений спеціально для мови програмування Python. Він використовує Python для визначення маршрутів, обробки запитів та реалізації функціональності веб-додатків.

Lighttpd (також відомий як "Lighty") - це легкий, швидкий та ефективний веб-сервер, який призначений для обробки статичних та динамічних веб-запитів. Він є відкритим програмним забезпеченням і є популярним вибором для хостинг-провайдерів, системних адміністраторів та розробників.

Далі розглянемо основні особливості Lighttpd:

- Ефективність: Lighttpd працює з низькими системними ресурсами, що дозволяє йому ефективно обробляти великі обсяги веб-трафіку з мінімальним використанням пам'яті та процесорного часу. Його архітектура оптимізована для високопродуктивної роботи з великою кількістю одночасних з'єднань.
- Підтримка FastCGI та SCGI: Lighttpd підтримує протоколи FastCGI (Fast Common Gateway Interface) та SCGI (Simple Common Gateway Interface), що дозволяє зв'язатися з зовнішніми процесами, які обробляють динамічний контент, такий як PHP, Python, Perl і багато інших. Це дозволяє використовувати Lighttpd як веб-сервер для виконання скриптів на різних мовах програмування.
- Розширені можливості налаштування: Lighttpd надає гнучкі

налаштування імпульсу, яке дозволяє змінювати його поведінку і параметри роботи в залежності від потреб проекту. Це включає налаштування віртуальних хостів, маршрутизацію URL, кешування, обробку компресії та багато іншого.

- Підтримка безпеки: Lighttpd має вбудовану підтримку TLS/SSL, що дозволяє шифрувати з'єднання між веб-сервером та клієнтом, забезпечуючи безпеку передачі даних. Він також має вбудовані функції обмеження доступу до файлів та обробки автентифікації, що допомагає забезпечити безпеку веб-додатків.
- Легкість використання та налаштування: Lighttpd має простий конфігураційний файл і логіку налаштування, що полегшує його використання та налаштування навіть для новачків. Він також має широку документацію та активну спільноту користувачів, яка надає підтримку та поради.

Lighttpd часто використовується у ситуаціях, коли вимагається швидкий та легкий веб-сервер, який може обробляти велику кількість запитів. Він є популярним вибором для хостинг-провайдерів, веб-розробників та системних адміністраторів, які цінують його продуктивність та гнучкість[12].

2.2 Вибір інструментів для створення локальної мережі

Dnsmasq - є легким і простим у використанні інструментом, який поєднує функціональність DNS (Domain Name System) та DHCP (Dynamic Host Configuration Protocol). Він широко використовується як локальний DNS-сервер та DHCP-сервер у домашніх мережах, офісах або невеликих комп'ютерних мережах. Ось деякі особливості та можливості dnsmasq:

DNS-резольвер: Dnsmasq дозволяє перетворювати доменні імена (наприклад, www.example.com) на відповідні IP-адреси, що дозволяє комп'ютерам знаходити та з'єднуватися з різними серверами та ресурсами в Інтернеті. Він зберігає локальний кеш DNS, що покращує швидкість

відповіді на запити DNS та зменшує навантаження на зовнішні DNS-сервери.

DHCP-сервер: Dnsmasq може виступати в якості DHCP-сервера, який надає автоматичну конфігурацію мережевих параметрів для підключених пристроїв. Він може виділяти IP-адреси, підмаски, шлюзи, DNS-сервери та інші параметри для підключених клієнтів, спрощуючи процес налаштування мережі.

DNS-блокування реклами: Dnsmasq може бути налаштований для блокування рекламних доменних імен, що дозволяє фільтрувати небажані рекламні ресурси на рівні DNS. Це забезпечує покращену безпеку, швидкість та зручність при перегляді веб-сторінок.

Перенаправлення імен: Dnsmasq може перенаправляти запити DNS на локальні IP-адреси або на інші сервери, дозволяючи налаштовувати поведінку DNS-резольвера відповідно до потреб користувача.

Простий у використанні та налаштуванні: Dnsmasq має простий текстовий файл конфігурації, де можна вказати параметри DNS і DHCP. Він має також добре задокументовану синтаксичну структуру конфігураційного файлу та легкий у використанні командний рядок.

Розширені можливості: Dnsmasq підтримує багато додаткових функцій, таких як фільтрація запитів DNS за допомогою власних правил, автоматична реєстрація імен хостів в DNS, підтримка IPv6 та багато іншого.

Узагальнюючи, dnsmasq є потужним і легким інструментом для налаштування локального DNS-сервера та DHCP-сервера. Він надає функціональність DNS-резольвера, DHCP-сервера та додаткових можливостей, які дозволяють забезпечити швидке та надійне функціонування мережі[6].

Hostapd (Host Access Point Daemon) є програмним забезпеченням для створення точки доступу Wi-Fi на комп'ютері або одноплатному комп'ютері, такому як Raspberry Pi. Він дозволяє перетворити пристрій на бездротову точку доступу, до якої можна підключитися з інших пристроїв для обміну даними через Wi-Fi. Ось деякі особливості та можливості hostapd:

Створення точки доступу: Hostapd дозволяє налаштувати комп'ютер або одноплатний комп'ютер як бездротову точку доступу. Він створює нову Wi-Fi мережу, до якої можна підключатися з інших пристроїв, таких як смартфони, планшети або ноутбуки.

Захист мережі: Hostapd підтримує різні методи безпеки, такі як WPA (Wi-Fi Protected Access) та WPA2, що дозволяють зашифровувати з'єднання Wi-Fi та забезпечити безпеку передачі даних між точкою доступу і підключеними пристроями. Ви можете встановити пароль або використовувати інші методи аутентифікації для обмеження доступу до мережі.

Керування налаштуваннями мережі: Hostapd дозволяє налаштовувати різні параметри мережі, такі як ім'я мережі (SSID), канал, швидкість передачі даних, обмеження кількості підключених пристроїв та інші. Це дозволяє забезпечити належну роботу мережі згідно зі специфічними потребами та вимогами.

Підтримка режиму моста: Hostapd може бути налаштований для роботи в режимі моста (bridge mode), що дозволяє об'єднати бездротову мережу з проводовою мережею. Це особливо корисно, коли потрібно розширити зону покриття мережі або підключити бездротові пристрої до існуючої проводової мережі.

Розширені налаштування: Hostapd має багато додаткових налаштувань, що дозволяють налаштовувати більш детальні параметри точки доступу, такі як використання мережі IEEE 802.11n/ac, управління потоком даних, контроль потужності передачі та інші.

Узагальнюючи, Hostapd є потужним інструментом для створення бездротової точки доступу з використанням комп'ютера або одноплатного комп'ютера. Він надає можливість налаштування мережі Wi-Fi, забезпечення безпеки передачі даних та розширені налаштування для належного управління точкою доступу[5].

2.3 Вибір мов програмування для розробки веб-інтерфейсу

Опис мов програмування:

PHP (Hypertext Preprocessor) - це скриптова мова програмування загального призначення, яка широко використовується для розробки веб-додатків. Ось кілька ключових рис мови програмування PHP:

Синтаксис: Синтаксис PHP схожий на синтаксис C, що робить його зрозумілим і легким для освоєння для багатьох програмістів. Код PHP вбудовується безпосередньо в HTML-сторінки та інтерпретується на сервері.

Веб-розробка: PHP був розроблений спеціально для веб-розробки. Він має вбудовану підтримку для створення динамічного веб-контенту, обробки форм, доступу до баз даних, керування сеансами користувачів та іншої функціональності, необхідної для розробки веб-додатків.

Підтримка баз даних: PHP має вбудовану підтримку для багатьох популярних систем керування базами даних, таких як MySQL, PostgreSQL, SQLite і багато інших. Це дозволяє розробникам легко працювати з базами даних і виконувати операції вставки, оновлення, видалення та вибірки даних.

Розширюваність: PHP має широкий набір вбудованих функцій і бібліотек, а також підтримує можливість додавання власних розширень та сторонніх бібліотек. Це дозволяє розробникам розширювати можливості мови і використовувати готові рішення для швидкого розробки.

Переносимість: PHP є переносимою мовою, що означає, що веб-додатки, написані на PHP, можуть працювати на різних операційних системах, таких як Windows, Linux, macOS і багато інших. Це забезпечує гнучкість і доступність для розробників.

Активна спільнота: PHP має велику та активну спільноту розробників, яка постійно працює над покращеннями мови, розробкою нових функцій і вирішенням проблем безпеки. Існує багато ресурсів, документації, форумів та бібліотек, які допомагають розробникам отримати підтримку та розв'язати свої завдання.

PHP є однією з найпопулярніших мов програмування для розробки веб-додатків, і вона використовується великою кількістю веб-сайтів та проєктів. Вона пропонує зручний і ефективний спосіб створення динамічних інтерактивних веб-додатків[1].

Python - це високорівнева, інтерпретована мова програмування загального призначення. Ось кілька ключових рис мови програмування Python:

Синтаксис і читабельність: Синтаксис Python відзначається своєю простотою та читабельністю. Вона використовує відступи для визначення блоків коду, що полегшує розуміння структури програми. Python прагне мати чистий і елегантний синтаксис, що сприяє швидкому розвитку програм.

Широкий спектр застосувань: Python є мовою загального призначення і знайшов своє застосування в багатьох галузях, таких як веб-розробка, наукові обчислення, штучний інтелект, автоматизація, аналітика даних, ігрова розробка та багато інших. Вона має велику кількість бібліотек і фреймворків, які полегшують розробку проєктів у різних сферах.

Простота використання: Python відомий своєю простотою використання. Вона має простий і зрозумілий синтаксис, що дозволяє розробникам швидко створювати програми. Python також підтримує динамічну типізацію, що означає, що ви не повинні передбачати тип змінної при її визначенні.

Багата бібліотека: Python має велику кількість стандартних бібліотек, що розширюють функціональність мови. Вони включають модулі для роботи з рядками, мережами, файлами, базами даних, графікою, обробки зображень, наукових обчислень та багато іншого. Крім того, існує також велика кількість сторонніх бібліотек, які розширюють можливості Python і дозволяють легко виконувати специфічні завдання.

Переносимість: Python підтримує багато платформ, включаючи Windows, macOS, Linux і багато інших. Це означає, що програми, написані на Python, можуть працювати на різних операційних системах без необхідності

внесення значних змін.

Розширюваність: Python є мовою, яка легко розширюється. Ви можете використовувати код, написаний на C або C++, для покращення продуктивності ваших програм або для використання існуючих бібліотек, які були розроблені на мовах низького рівня.

Python є однією з найпопулярніших мов програмування в світі завдяки своїм перевагам у простоті, читабельності та розширюваності. Вона відмінно підходить для початківців, але також є потужним інструментом для професійних розробників[3].

JavaScript - це високорівнева мова програмування, яка використовується для розробки веб-додатків. Ось кілька ключових рис мови програмування JavaScript:

Веб-розробка: JavaScript є основною мовою для розробки веб-додатків. Вона використовується для створення інтерактивних елементів на веб-сторінках, контролю поведінки веб-сторінок, обробки подій, маніпуляції DOM (Document Object Model) та взаємодії з сервером.

Клієнтська і серверна сторони: JavaScript може виконуватися як на стороні клієнта (в браузері) так і на стороні сервера (з використанням платформ, таких як Node.js). Це дозволяє розробникам створювати повноцінні веб-додатки, які працюють як на клієнтському, так і на серверному рівні.

Динамічна типізація: JavaScript є мовою з динамічною типізацією, що означає, що типи змінних визначаються автоматично під час виконання програми. Це надає більшу гнучкість і зручність у роботі з даними.

Об'єктно-орієнтований підхід: JavaScript підтримує об'єктно-орієнтоване програмування, що дозволяє організовувати код у вигляді об'єктів, які мають свої властивості та методи. Це полегшує організацію та підтримку складних проектів.

Багата функціональність: JavaScript має велику кількість вбудованих функцій і методів, які дозволяють розробникам взаємодіяти з елементами

веб-сторінок, виконувати математичні операції, робити запити до сервера, маніпулювати рядками, масивами, об'єктами та іншими типами даних.

Розширюваність: JavaScript підтримує використання сторонніх бібліотек і фреймворків, що дозволяє розробникам використовувати готові рішення для швидкого розробки додатків. Наприклад, є велика кількість бібліотек для роботи з візуалізацією даних, створення анімацій, роботи з AJAX-запитами, реалізації шаблонів та багато іншого.

JavaScript є важливою мовою програмування для веб-розробки і використовується мільйонами розробників по всьому світу. Вона надає потужні можливості для створення динамічних веб-додатків, забезпечує взаємодію з користувачем та реалізує різноманітні функціональні можливості[9].

Опис мов web розмітки: HTML (Hypertext Markup Language) - це стандартна мова розмітки для створення веб-сторінок. Ось кілька ключових рис мови HTML:

Структура сторінки: HTML дозволяє визначати структуру веб-сторінки за допомогою тегів. Теги вказують браузеру, як правильно відображати різні елементи, такі як заголовки, абзаци, списки, таблиці, зображення та посилання.

Текстовий контент: HTML надає можливість вставляти текстовий контент на веб-сторінку. Він підтримує різні рівні заголовків (h1-h6) для ієрархічного представлення тексту, абзаци, списки (нумеровані та марковані), таблиці для відображення даних, та інші елементи форматування, такі як жирний текст, курсив, підкреслення, виноска, заголовки, цитати та інші.

Зображення та мультимедіа: HTML дозволяє вставляти зображення на веб-сторінку за допомогою тегу <img>. Крім того, він підтримує вставку відео- та аудіофайлів за допомогою тегів <video> та <audio>. Це дозволяє розширити можливості сторінки та забезпечити мультимедійний контент.

Гіперпосилання: HTML надає засоби для створення гіперпосилань на інші веб-сторінки або на різні частини поточної сторінки. Це здатність, що

дозволяє навігувати між різними сторінками та ресурсами в Інтернеті.

Форми: HTML дозволяє створювати веб-форми для збору даних від користувачів. Форми можуть містити текcові поля, кнопки, перемикачі, прапорці, список вибору, календарі та інші елементи, що дозволяють збирати інформацію від користувача.

Метадані: HTML також надає можливість визначати метатеги, які надають додаткову інформацію про веб-сторінку. Наприклад, метатеги можуть вказувати мову сторінки, кодування символів, автора, ключові слова для пошукової оптимізації та інші метадані.

HTML використовується разом з CSS (Cascading Style Sheets) та JavaScript для створення веб-сторінок з багатим функціоналом та стильовим оформленням. Він є основою веб-розробки та дозволяє розробникам створювати сторінки, які здатні відтворюватися на різних браузерах і пристроях[7].

CSS (Cascading Style Sheets) - це мова програмування, яка використовується для опису зовнішнього вигляду веб-сторінок, стильового оформлення та макетування. Ось кілька ключових рис мови CSS:

Відокремлення стилю від вмісту: CSS дозволяє розділити стильове оформлення веб-сторінки від її структури та вмісту. Це полегшує редагування та зміну вигляду сторінки, оскільки стилі можуть бути застосовані до багатьох елементів одночасно.

Властивості та значення: CSS надає широкий набір властивостей, які визначають різні аспекти стилю елементів веб-сторінки. Наприклад, властивості можуть визначати кольори, шрифти, розміри, відступи, рамки, фонові зображення та багато іншого. Для кожної властивості в CSS вказується відповідне значення, яке визначає, як саме ця властивість буде застосовуватися.

Каскадність і спадковість: CSS працює на основі принципу каскаду, що дозволяє визначати багато рівнів стилів і встановлювати пріоритети між ними. Крім того, CSS використовує спадковість, що означає, що стилі, задані

для батьківського елемента, також можуть застосовуватися до його дочірніх елементів.

Класи та ідентифікатори: CSS дозволяє використовувати класи і ідентифікатори для точного вибору і стилізації певних елементів на сторінці. Класи використовуються для групування елементів з однаковими стилями, тоді як ідентифікатори використовуються для унікальної ідентифікації конкретного елемента.

Респонсивний дизайн: CSS дозволяє створювати адаптивні та респонсивні веб-сторінки, які коректно відображаються на різних пристроях та розмірах екранів. За допомогою медіа-запитів і флексібільних розмірів, можна змінювати розташування елементів та їх стиль в залежності від розміру екрану[8].

CSS використовується разом з HTML та JavaScript для створення зовнішнього вигляду веб-сторінок. Він дозволяє розробникам контролювати вигляд, макет та стиль елементів на сторінці, створюючи привабливий та професійний дизайн.

3 ТЕОРЕТИЧНИЙ АНАЛІЗ ЛОКАЛЬНИХ КОМП’ЮТЕРНИХ МЕРЕЖ

Комп'ютерна мережа включає в себе комп'ютери та інші пристрої, які з'єднані між собою для передачі даних. Зазвичай розрізняють два типи комп'ютерних мереж: локальні (LAN) і глобальні (WAN).

Локальна мережа (LAN) - це мережа, в якій комп'ютери розташовані на невеликій відстані один від одного (зазвичай до 1-2 км) і не використовують загальнодоступні засоби зв'язку, такі як телефонні мережі. Локальна мережа може належати одній організації і зазвичай використовує дорогі високоякісні лінії зв'язку, що дозволяють досягати високих швидкостей передачі даних до 100 Мбіт/с.

Локальні обчислювальні мережі можна розділити на два основні типи: однорівневі (Peer to Peer) і ієрархічні (мережі з виділеним сервером).

У однорівневій мережі комп'ютери є рівноправними, кожен має унікальне ім'я та, зазвичай, пароль для входу. Ім'я та пароль призначаються власником комп'ютера через операційну систему. Однорівневі мережі мають переваги низької вартості і високої надійності, але можуть мати обмеження у продуктивності, складності управління, захисті інформації та оновленні програмного забезпечення.

В ієрархічних мережах є один або кілька спеціальних комп'ютерів - серверів, на яких зберігається спільна інформація для користувачів. Взаємодія між робочими станціями відбувається через ці сервери. Сервери зазвичай є потужними комп'ютерами з великою кількістю ресурсів, і їх використовують для надійного зберігання та обміну даними. Мережі з виділеним сервером мають переваги надійності, високої швидкодії, простоти управління та можуть обслуговувати більше робочих станцій, але вимагають витрат на виділення сервера і можуть бути менш гнучкими.

Локальні мережі також можуть бути класифіковані за призначенням, швидкістю передачі, типом доступу та фізичним середовищем передачі. Деякі менш поширені класифікації включають кільцеві, шинні, зіркоподібні, деревоподібні мережі, а також враховують швидкість передачі даних і тип

методу доступу. На рисунку 2.1 показані фізичні топології локальної мережі.

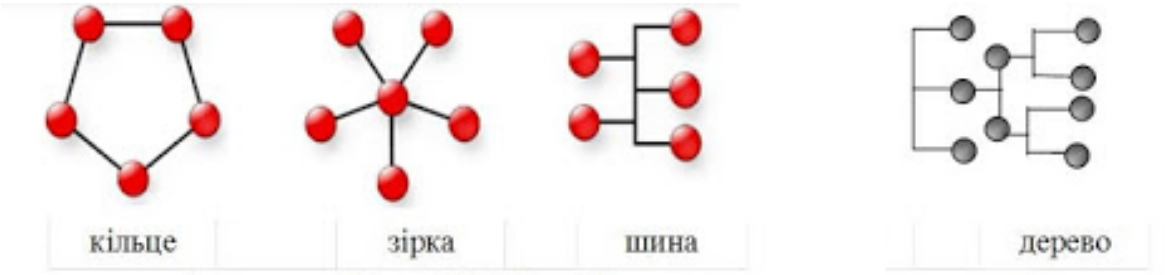


Рисунок 2.1 - Фізичні топології локальної мережі

3.1 Топологія шина

Мережова топологія "Шина" є однією з основних фізичних топологій для локальних обчислювальних мереж. В цій топології всі комп'ютери підключені до одного спільного провідника, який називається "шиною". У топології "Шина" кожен комп'ютер має доступ до цієї спільної шини. На обох кінцях шини розташовані спеціальні пристрої, які називаються термінаторами. Вони відповідають за узгоджування проходження даних по шині.

Основною перевагою топології "Шина" є її простота в монтажі та низька вартість. Вона не вимагає складного кабельного з'єднання і дозволяє легко додавати нові комп'ютери до мережі. Крім того, вона може бути ефективно використана для невеликих мереж.

Проте, топологія "Шина" має певні недоліки. Одним з них є проблематичність локалізації місця несправностей. Якщо кабель, що утворює шину, пошкоджується у будь-якому місці, це може призвести до припинення обміну інформацією між всіма комп'ютерами в мережі. Крім того, низька надійність є ще одним обмеженням топології "Шина". Якщо шина пошкоджується або розривається, то це також призведе до втрати зв'язку між усіма комп'ютерами.

Також важливо враховувати особливості поширення електричного сигналу. Навіть якщо два комп'ютери фізично з'єднані один з одним, якщо на

одному кінці шини відсутній термінатор, зв'язок між ними буде неможливим через властивості поширення сигналу по шині.

3.2 Топологія зірка

Мережова топологія "Зірка" є однією з найпоширеніших фізичних топологій для локальних обчислювальних мереж. В цій топології кожен комп'ютер підключений окремим кабелем до центрального пристрою, який називається комутатором або концентратором.

У топології "Зірка" всі кабелі радіально виходять з комутатора і з'єднуються з кожним комп'ютером в мережі. Комутатор відповідає за пересилання даних між комп'ютерами, відправляючи і отримуючи пакети інформації.

Основною перевагою топології "Зірка" є її надійність і простота управління. У разі відмови або пошкодження одного з кабелів, це не впливає на решту мережі, і комп'ютери можуть продовжувати працювати. Також, додавання нових комп'ютерів до мережі або видалення старих є простим завдяки централізованій структурі. Крім того, топологія "Зірка" забезпечує високу швидкодію передачі даних. Кожен комп'ютер має власний кабель, що дозволяє уникнути колізій або конфліктів при передачі інформації.

Проте, недоліком топології "Зірка" є залежність від роботи комутатора. Якщо комутатор вийде з ладу або перестане працювати, всі комп'ютери, підключені до нього, втратять зв'язок. Тому, з метою забезпечення високої доступності, можуть використовуватися резервні комутатори або створюватися додаткові шляхи з'єднання. Також важливо враховувати обмеження на відстань між комп'ютерами і комутатором. Велика відстань може призвести до зниження якості сигналу і швидкості передачі даних.

Взагалі, топологія "Зірка" є надійним і ефективним варіантом для будівництва локальних обчислювальних мереж, зокрема в офісних середовищах або невеликих комп'ютерних мережах.

3.3 Топологія кільце

Мережова топологія "Кільце" є однією з основних фізичних топологій для локальних обчислювальних мереж. У цій топології комп'ютери підключені до мережі у вигляді кільця, де кожен комп'ютер має пряме з'єднання з двома сусідніми комп'ютерами.

У топології "Кільце" дані передаються від одного комп'ютера до наступного у формі кільцевого потоку. Кожен комп'ютер функціонує як повторювач (repeater), що підсилює сигнал і передає його наступному комп'ютеру. Цей процес продовжується до отримання даних необхідним комп'ютером.

Однією з переваг топології "Кільце" є висока стійкість до відмов. Якщо один з комп'ютерів або сегментів кільця вийде з ладу, інформація все одно може продовжувати курсувати в іншому напрямку, оминаючи неполадку. Це забезпечує надійність мережі навіть у випадку відмови окремих вузлів. Крім того, топологія "Кільце" відома своєю ефективністю. У разі передачі даних по кільцю виникає мінімальна кількість колізій або конфліктів, оскільки кожен комп'ютер може передавати дані тільки у своєму напрямку. Це дозволяє досягти високої швидкодії передачі даних у мережі.

Однак, існує певний недолік у топології "Кільце". Якщо виникає несправність на кільці, то вона може призвести до переривання зв'язку в усій мережі. Ремонт та виявлення місця несправності можуть бути викликані складністю структури кільця. Також, збільшення масштабу мережі "Кільце" може призвести до зниження продуктивності через збільшення часу передачі даних по кільцю.

3.4 Топологія Дерево

Мережова топологія "Дерево" є ієрархічною структурою, яка базується на принципі розгалуження подібно до дерева. У цій топології комп'ютери

організовані у вигляді деревоподібної мережі, де вузли вузли розгалужуються на підвузли, а ті, у свою чергу, нащадків.

Топологія "Дерево" має центральний вузол, відомий як кореневий вузол або корінь. Всі інші вузли підключені до кореня шляхом послідовного розгалуження. Кожен вузол може мати кілька підвузлів, а підвузли можуть мати свої власні підвузли. Цей процес розгалуження може продовжуватися на декілька рівнів, утворюючи ієрархічну структуру. Однією з переваг топології "Дерево" є висока масштабованість та ефективне управління мережею. Шляхом розгалуження можна підключати нові вузли та розподіляти навантаження між різними рівнями мережі. Це дозволяє створювати великі мережі з багатьма комп'ютерами, підвищуючи їх продуктивність та надійність.

Також, топологія "Дерево" забезпечує високий рівень безпеки та контролю доступу до ресурсів. Кореневий вузол може встановлювати правила та обмеження для підвузлів, контролювати доступ до даних та регулювати мережевий трафік. Це дозволяє забезпечити захист мережі від несанкціонованого доступу та збереження конфіденційності даних.

Однак, топологія "Дерево" може мати певні обмеження щодо масштабування та надійності. В разі відмови кореневого вузла може бути порушена доступність всієї мережі. Крім того, додавання нових вузлів або зміна структури мережі може вимагати значних зусиль та часу.

Узагальнюючи, топологія "Дерево" є ефективним рішенням для середніх та великих мереж, де важливі масштабованість, управління та безпека. Вона забезпечує ієрархічну структуру, що дозволяє розподіляти навантаження та керувати доступом до ресурсів, що робить її популярним вибором для багатьох організацій та підприємств.

Мережеві топології відіграють важливу роль у впорядкуванні та організації комп'ютерних мереж. Кожна топологія має свої переваги та обмеження, які варто враховувати при виборі найбільш підходящого варіанту для конкретного середовища.

Топологія "Шина" відрізняється простотою та низькими витратами, але може мати проблеми з надійністю та локалізацією несправностей.

Топологія "Зірка" забезпечує високий рівень надійності, легкість відновлення та керування мережею, але вимагає більшої кількості кабелів та активного обладнання.

Топологія "Кільце" має високу стійкість до відмов та ефективне використання пропускної здатності, але може страждати від відмови одного вузла.

Топологія "Дерево" забезпечує ієрархічну структуру, масштабованість та керування, але може бути обмежена у випадку відмови кореневого вузла.

Крім цих основних топологій, існує також комбінація різних топологій, яка називається гібридною топологією. Гібридна топологія дозволяє поєднувати переваги різних топологій та використовувати їх відповідно до потреб конкретної мережі. При виборі мережевої топології варто враховувати такі фактори, як масштаб мережі, надійність, пропускна здатність, безпека, легкість керування та вартість реалізації. Кожна топологія має свої особливості та відповідає різним потребам та сценаріям застосування. Знання про різні мережеві топології допомагає інженерам та адміністраторам при проектуванні, розгортанні та управлінні комп'ютерними мережами, забезпечуючи оптимальне функціонування, надійність та ефективність зв'язку між комп'ютерами та пристроями.

Мережева технологія представляє собою комплексний набір стандартних протоколів та апаратно-програмних засобів, включаючи мережеві адаптери, драйвери, кабелі, роз'єми і т. д. Цей набір забезпечує достатній функціонал для побудови комп'ютерної мережі. Нижче наведено найбільш відомі мережеві технології разом з їх основними характеристиками:

Технологія ARCnet:

- Логічна топологія - шина.
- Фізична топологія - шина, зірка, змішана.
- Середовище передачі сигналу - коаксіальний кабель (93 Ом), кручена

пара.

- Швидкість обміну інформацією - 2,5 Мбіт / сек.
- Максимальна довжина з'єднань - від 100 до 610 метрів (в залежності від типу з'єднувача).

Максимальна кількість вузлів в одній мережі - 255. Максимальний розмір мережі (сумарна довжина з'єднань) - 6000 метрів.

В даний час апаратура для мереж ARCnet не випускається.

100VG-AnyLAN:

- Логічна топологія - дерево (різновид зірки).
- Фізична топологія - дерево (різновид зірки).
- Середовище передачі сигналу - кручена пара (обов'язково чотирипарного).
- Швидкість обміну інформацією - 100 Мбіт / сек.
- Максимальна довжина з'єднань - від 100 до 200 метрів (в залежності від типу з'єднувача).
- Максимальна кількість вузлів в одній мережі - +1024.
- Максимальний розмір мережі - 2000 метрів.

Апаратура для організації локальних обчислювальних мереж за технологією 100VG-AnyLAN випускається практично тільки фірмою Hewlett-Packard, вартість її досить висока, тому дана технологія не набула поширення.

Token Ring:

- Логічна топологія - кільце.
- Фізична топологія - зірка.
- Середовище передачі сигналу - вита пара, волоконно-оптичний кабель.
- Швидкість обміну інформацією - 4; 16; 100 і 1000 Мбіт / сек.
- Максимальна довжина з'єднань - від 100 до10000 метрів (в залежності від типу з'єднувача).
- Максимальна кількість вузлів в одній мережі - до 260 (в залежності від типу з'єднувача).

Ціна - висока, що різко звужує сферу застосування, принаймні, в нашій країні. В іншому світі технологія Token Ring поряд з технологією Ethernet є однією з найбільш поширених.

FDDI:

- Логічна топологія - кільце.
- Фізична топологія - кільце, зірка або їх гібриди.
- Середовище передачі сигналу - волоконно-оптичний кабель.
- Швидкість обміну інформацією - 100 Мбіт / сек.
- Максимальна довжина з'єднань - від 2 до 60 кілометрів (залежно від типу волоконно-оптичного кабелю).
- Максимальна кількість вузлів в одній мережі - 500.
- Максимальна загальна довжина мережі - до 200 км.

Ціна - висока. Незважаючи на те, що технологія FDDI розроблялася для локальних обчислювальних мереж, з огляду на вартість область її застосування - мережі міського масштабу і більші.

Ethernet:

- Логічна топологія - шина.
- Фізична топологія - шина, зірка.

Для передачі сигналу по провідникам необхідний певний час, від джерела до приймача, який, хоч і непомітний для людини, все ж потрібний. У деяких випадках можуть виникати ситуації, коли один абонент мережі починає передавати інформацію, не знаючи про передачу іншим абонентом, що призводить до колізії - "зіткнення" пакетів даних. Перший абонент, виявивши колізію, повідомляє про неї всю мережу. Всі абоненти припиняють передачу і чекають випадковий проміжок часу перед тим, як знову спробувати передати дані. Важливо, щоб колізія була виявлена до закінчення передачі будь-яким абонентом.

Для мереж Ethernet, які використовують виту пару, важливим є правило "чотирьох хабів", яке стверджує, що між будь-якими двома абонентами мережі не повинно бути більше ніж чотири хаби. При дотриманні цього

правила, а також обмеження на довжину з'єднувального кабелю, будь-яка колізія, яка виникне, буде виявлена і оброблена учасниками процесу передачі інформації.

Існує безліч протоколів, які беруть участь у взаємодії мережі. Кожен протокол має свої цілі, виконує різні завдання і має власні переваги і обмеження.

Протоколи працюють на різних рівнях моделі взаємодії відкритих систем OSI/ISO. Функції протоколів залежать від рівня, на якому вони працюють. Кілька протоколів можуть співпрацювати разом, утворюючи стек або набір протоколів. Протоколи розподілені по різних рівнях стека протоколів, що відповідають рівням моделі OSI. Загалом, протоколи надають повну характеристику функцій і можливостей стека протоколів.

Давайте розглянемо найбільш поширені протоколи передачі даних, що використовуються в даний час.

Протокол управління передачею даних (Transmission Control Protocol/Internet Protocol - TCP/IP) став прийнятою мовою в комп'ютерному світі, що використовується як основа для мережі Інтернет. Більшість розробників мережевих операційних систем, таких як Windows 2000 Server, Novell NetWare 5.x, UNIX і Linux, визнали TCP/IP як основний мережевий протокол за замовчуванням.

Набір протоколів TCP/IP включає різні "учасники", які працюють разом. Цей стек протоколів TCP/IP складається з чотирьох рівнів:

- Прикладний рівень (Application Layer): протоколи, такі як HTTP, RTSP, FTP, DNS.
- Транспортний рівень (Transport Layer): протоколи, такі як TCP, UDP, SCTP, DCCP.
- Мережевий і міжмережевий рівень (Internet Layer): протокол IP (додаткові протоколи, наприклад ICMP і IGMP, працюють на рівні мережі; протокол ARP є самостійним протоколом, який працює на канальному рівні).

- Канальний рівень (Network Access Layer): технології Ethernet, IEEE 802.11 WLAN, SLIP, Token Ring, ATM і MPLS, фізичне середовище і принципи кодування інформації, T1, E1.

Давайте розглянемо основні протоколи передачі даних системи TCP/IP:

- Транспортні протоколи: TCP (Transmission Control Protocol) та інші, відповідають за керування передачею даних між комп'ютерами.
- Протоколи маршрутизації: IP (Internet Protocol) та інші, забезпечують фактичну передачу даних, визначають найкращий шлях до адресата.
- Протоколи підтримки мережевої адреси: DNS (Domain Name System) та інші, визначають унікальну адресу комп'ютера.
- Протоколи прикладних сервісів: FTP (File Transfer Protocol), HTTP (HyperText Transfer Protocol), TELNET та інші, використовуються для передачі файлів, доступу до WWW, віддаленого термінального доступу і т.д.
- Шлюзові протоколи: EGP (Exterior Gateway Protocol) та інші, допомагають передавати повідомлення про маршрутизацію і стан мережі, а також обробляти дані для локальних мереж.
- Поштові протоколи: POP (Post Office Protocol) - використовується для прийому електронної пошти, SMTP (Simple Mail Transfer Protocol) - використовується для передачі поштових повідомлень.

IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange) - це набір мережевих протоколів, розроблений компанією Novell для мереж на базі операційної системи Novell NetWare. Цей набір протоколів забезпечував функціонування файлових серверів і серверів друку в локальних мережах з 1980-х років. Хоча IPX/SPX і TCP/IP є наборами протоколів з різними функціями для передачі даних в мережі, IPX/SPX став менш популярним і використовується переважно в старих мережах, тоді як TCP/IP є основним мережевим протоколом.

На сьогоднішній день TCP/IP є основним мережевим протоколом за замовчуванням, а інші протоколи менш поширені в локальних мережах.

3.5 Канали та фізичні пристрої мережевого зв’язку

У локальних мережах існують різні типи каналів зв'язку, які є фізичним середовищем передачі інформації і важливою складовою мережі. Однією з основних характеристик каналу зв'язку є його пропускна здатність, тобто максимальна швидкість передачі даних. В локальних мережах використовуються наступні типи каналів зв'язку:

Вита пара (скручена пара): Це найпоширеніший і доступний тип кабелю. Максимальна відстань передачі становить 1,5-2,0 кілометра, а максимальна швидкість - 1,2 Гбіт/с. Вона має менший захист від завад порівняно з коаксіальним кабелем. Тривалість поширення сигналу становить 8-12 наносекунд на метр, а термін експлуатації - 2-6 років. Вита пара зараз є основним засобом передачі даних у локальних мережах.

Коаксіальний кабель: Цей тип кабелю є одним з найпоширеніших для передачі даних у локальних мережах поряд з витою парою. Він має високу швидкість передачі, високу стійкість до завад, довговічність і помірну вартість. Для підключення до локальних мереж використовуються спеціальні засоби з'єднання. Коаксіальний кабель використовується для з'єднання комп'ютерів в топології шина, що є простим способом без потреби додаткового обладнання. Термін експлуатації такого кабелю становить 10-12 років.

Волоконно-оптичний (оптоволоконний) кабель: Використовуються прозорі скляні волокна як фізичне середовище для передачі сигналів. Швидкість передачі даних по такому кабелю становить 0,2-1,0 Гбіт/с, а максимальна теоретична швидкість - 200 Гбіт/с. Можлива довжина сполучень до 110 кілометрів. Волоконно-оптичний кабель має менше загасання сигналу порівняно з коаксіальним кабелем, вищу швидкість передачі, широку частотну смугу і невразливість до електромагнітних завад. Проте він має обмежену механічну стійкість, не може бути гнутий, тертий, пересуватись або витримувати вібрації.

Існують також бездротові локальні мережі, де передача інформації між комп'ютерами здійснюється через використання високочастотного діапазону (НВЧ) або інфрачервоних променів.

Один з пристроїв мережевого зв'язку, який використовується для об'єднання комп'ютерів у невеликих або тимчасових локальних мережах, є концентратор. Інші пристрої, такі як репітери, комутатори і маршрутизатори, відносяться до пристроїв мережевого обміну. Вони використовуються для забезпечення обміну даними в мережі. Міжмережевий обмін відбувається в об'єднаних локальних мережах, коли застосовується спеціальна технологія для розширення звичайних меж локальної мережі або для об'єднання кількох локальних мереж в єдину велику мережу.

Концентратори, також відомі як хаби (Hub), є найпоширенішими з'єднувальними пристроями в локальних мережах. Однак, в останні часи їх швидко витісняють більш економічні комутатори. У локальних мережах концентратори виступають центральними з'єднувальними вузлами. Зазвичай вони не мають активних електронних схем, тому не можуть бути використані для розширення мережі. Замість цього, їх основна функція полягає в з'єднанні кабелів та передачі інформаційних сигналів між усіма комп'ютерами в мережі. Основний принцип роботи концентратора полягає у трансляції пакетів даних, які надходять на один з його портів, на всі інші порти одночасно.

Концентратори застосовуються в мережах, де використовується кручена пара. Порти концентратора служать точками з'єднання для мережевих пристроїв. Комп'ютери та інші пристрої підключаються до концентратора за допомогою окремих кабелів. Концентратори можуть відрізнятися за формою, розміром і кількістю портів. Якщо кількості портів концентратора недостатньо, можна підключити ще один концентратор (їх можна з'єднати в "гірлянду" за допомогою короткого сполучного кабелю).

Концентратори мають різноманітні форми і розміри, а також значно варіюються у ціні. Зазвичай, чим більше портів має концентратор, тим вища

його вартість. Концентратори, які підтримують швидкі варіанти Ethernet, такі як Fast Ethernet, мають більш високу ціну.

Репітери, також відомі як повторювачі, використовуються, коли розмір локальної мережі перевищує максимальну довжину кабелю, що використовується. Репітер відновлює сигнали, отримані від комп'ютерів та інших мережевих пристроїв, зберігаючи цілісність сигналу на більшій відстані, ніж дозволяють мережеві кабелі.

Репітери не здатні маршрутизувати мережевий трафік або визначати шлях передачі даних; вони є простими пристроями, які лише підсилюють отриманий сигнал. Недоліком репітерів є те, що вони підсилюють не лише сигнал, але й перешкоди. У гірших випадках шум у лінії може зробити передаваний потік даних незрозумілим.

Міст (Bridge) - це пристрій мережевого обміну, який дозволяє зберігати пропускну здатність в мережі. При зростанні локальної мережі потік мережевої інформації може перевищити пропускну здатність мережного середовища. Одним із способів збереження пропускної здатності в мережі є розбиття її на сегменти, які з'єднані за допомогою мостів. Мости володіють більшими можливостями ніж концентратори і репітери, вони навіть використовують спеціальне програмне забезпечення.

Міст може прочитати MAC-адреси (відомі також як апаратні адреси, які знаходяться в пам'яті мережевої карти кожного комп'ютера в мережі) у кожному пакеті даних, що циркулюють по сегментах мережі, які підключені до мосту. Знаючи MAC-адреси у кожному з сегментів мережі, міст перешкоджає передачі даних, що належать одному сегменту, до інших сегментів мережі, які не обслуговуються ним.

Комутатор, також відомий як свіч (Switch), є ще одним пристроєм мережевого обміну, який використовується для керування пропускною здатністю великої мережі. Використання комутаторів стає все більш поширеним, навіть у невеликих мережах, оскільки вони дозволяють ефективно керувати використанням пропускної спроможності.

Комутатор управляє потоком даних за допомогою MAC-адреси, яка міститься в кожному пакеті даних (вона збігається з MAC-адресою мережевої карти комп'ютера). Комутатори об'єднують мережі в віртуальні локальні мережі (Virtual Local Area Network - VLAN). Головна перевага VLAN полягає в тому, що вона логічно об'єднує комп'ютери в комунікаційні групи, незалежно від їх фізичного розташування. Це означає, що комп'ютери, які обслуговують один тип користувачів, можуть бути частиною однієї віртуальної локальної мережі зі спільною пропускною здатністю. Наприклад, якщо інженери компанії розташовані у різних частинах офісної будівлі, їх комп'ютери можуть належати до однієї віртуальної мережі зі спільним ресурсом пропускної здатності.

Програмне та апаратне забезпечення комутаторів використовуються для передачі пакетів між комп'ютерами та іншими мережевими пристроями. Кожен комутатор має свою власну операційну систему. Зазвичай можна переглянути апаратну (MAC) адресу та IP-адресу комутатора. Інші статистичні дані включають кількість відправлених і отриманих пакетів.

Маршрутизатори (Router) пропонують ще більш широкі можливості порівняно з мостами і комутаторами (маршрутизатори діють на більш високому мережевому рівні - рівні концептуальної моделі OSI, ніж мости і комутатори, які працюють на рівні каналу передачі даних). Апаратне та програмне забезпечення маршрутизатора дозволяє ефективно маршрутизувати дані від джерела до призначення (під програмним забезпеченням мається на увазі операційна система). У маршрутизаторів використовується складна операційна система, яка надає можливість налаштування маршрутизації пакетів даних для різних мережевих протоколів, таких як TCP/IP, IPX/SPX і AppleTalk.

Маршрутизатори застосовуються для поділу великих та перенавантажених локальних мереж на сегменти, а також для з'єднання віддалених локальних мереж за допомогою різних технологій WAN [13].

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ HOT-SPOT на базі RASPBERRY PI

Елементи локальної мережі підприємства та їх IP-адресація.

Технічне завдання курсового проекту передбачає створення мережі підприємства, яке складається з восьми відділів:

1) Встановлення та налаштування OS Raspbian (Linux).
2) Встановлення та налаштування DHCP серверу.
3) Встановлення та налаштування конфігурацій для безпровідної точки доступа стандарта IEEE 802.11n/ac.
4) Встановлення та налаштування web серверу lighttpd .
5) Розробка web інтерфейсу “AirLink” для управлення апаратними .
6) Встановлення та алаштування системи IDS Snort для індетифікаціє атак на локальну мережу.
7) Розробка механізмів запобігання атак MITM на локальну мережу.

4.1 Встановлення та налаштування OS Raspbian

Raspbian - це операційна система, спеціально розроблена для одноплатних комп'ютерів Raspberry Pi. Вона є офіційною і рекомендованою операційною системою для Raspberry Pi Foundation.

Raspbian базується на операційній системі Debian, вона оптимізована для використання на низькопотужних пристроях Raspberry Pi. Ця ОС пропонує широкий набір функцій і інструментів, що дозволяють розробникам, студентам та ентузіастам з легкістю використовувати Raspberry Pi для різних проектів.

Основні особливості Raspbian:

Легка у використанні: Raspbian надає зручний інтерфейс користувача, що дозволяє легко навігувати по системі та налаштовувати параметри.

- Підтримка апаратного забезпечення Raspberry Pi: Raspbian повністю сумісний з апаратним забезпеченням Raspberry Pi, що дозволяє максимально використовувати можливості пристрою.

- Велика спільнота користувачів: Raspbian має велику спільноту користувачів і розробників, яка надає підтримку та допомогу у вирішенні питань та проблем.
- Набір попередньо встановлених програм: Raspbian поставляється з популярними програмами, такими як веб-браузер Chromium, текстовий редактор Nano, офісний пакет LibreOffice, медіаплеєр та інші, що дозволяє вам швидко почати роботу.
- Підтримка GPIO: Raspbian має вбудовану підтримку GPIO (загального призначення вводу-виводу), що дозволяє взаємодіяти з зовнішніми електронними компонентами і пристроями.
- Оновлення та підтримка: Raspbian отримує регулярні оновлення та підтримку з боку Raspberry Pi Foundation, що забезпечує безпеку і функціональність ОС.

Raspbian є популярним вибором для початківців, хобістів, освітніх установ та багатьох інших користувачів, які використовують Raspberry Pi для розробки різноманітних проектів.

Raspbian було обрано як оптимальну операційну систему для Raspberry Pi, оскільки вона була спеціально адаптована для пристрою. Raspbian є варіантом Debian, але з оптимізацією під апаратне забезпечення Raspberry Pi, що забезпечує кращу продуктивність. Інсталяція Raspbian також проста, оскільки доступні інсталяційні зображення, які легко записати на SD-карту та встановити на Raspberry Pi.

Офіційна підтримка та велика спільнота користувачів Raspberry Pi сприяють популярності Raspbian. Користувачі можуть знайти документацію, форуми та розширення, які полегшують використання та вирішення проблем. Крім того, Raspbian має доступ до широкого спектру програмного забезпечення, яке можна встановити на Raspberry Pi, розширюючи його можливості в різних сферах.

Базуючись на Debian, Raspbian також сумісний з багатьма програмами та пакетами Linux, що розроблені для Debian. Це дає користувачам доступ до

багатьох додатків та можливість використовувати свої улюблені операційні системи на Raspberry Pi.

Загалом, Raspbian було обрано як оптимальне рішення для Raspberry Pi через його оптимізацію, легкість встановлення, підтримку та широкий вибір програмного забезпечення.

Ядро (Kernel): Raspbian використовує спеціально зроблене для Raspberry Pi ядро Linux. Ядро відповідає за керування апаратним забезпеченням, включаючи взаємодію з процесором, пам'яттю, ввідно-вивідними пристроями та іншими компонентами Raspberry Pi.

Основна файлова система: Raspbian використовує розподілений файловий стандарт Linux (Filesystem Hierarchy Standard, FHS), де різні типи файлів розміщені у відповідних директоріях. Наприклад, конфігураційні файли зазвичай знаходяться у директорії /etc, програми - у /usr/bin, системні бібліотеки - у /usr/lib та інше.

Пакетний менеджер: Raspbian використовує Advanced Packaging Tool (APT) як пакетний менеджер. Він дозволяє користувачам встановлювати, оновлювати та керувати пакетами програмного забезпечення на Raspberry Pi. Користувачі можуть використовувати команду apt-get для взаємодії з APT і виконання різних операцій з пакетами.

Графічне середовище: Raspbian постачається з графічним середовищем, яким може бути LXDE (Lightweight X11 Desktop Environment), Xfce або PIXEL (Pi Improved Xwindows Environment, Lightweight). Графічне середовище надає користувачам зручний інтерфейс для взаємодії з операційною системою та запущеними програмами.

Набір програмного забезпечення: Raspbian містить набір програм та утиліт, які роблять його придатним для широкого спектру застосувань. Це включає текстовий редактор, веб-браузер, офісні програми, медіа-програвачі, програми для розробки, налаштування системи та багато іншого. Крім того, користувачі можуть встановлювати додаткове програмне забезпечення за допомогою пакетного менеджера APT або з інших джерел.

Додаткові компоненти: Raspbian також має спеціальні компоненти, пов'язані з Raspberry Pi, такі як інструменти для керування GPIO (General Purpose Input/Output) та іншими виводами-вводами, конфігураційні файлі для налаштування апаратного забезпечення Raspberry Pi, драйвери для підтримки специфічних функцій та інше.

Raspbian використовую архитектру Архітектура armhf (arm hard float) є варіантом архітектури ARM для систем з плаваючою комою (floating-point). Основна відмінність архітектури armhf від armel (arm little-endian) полягає в способі обробки дій з плаваючою комою.

Архітектура armhf використовує апаратну підтримку плаваючої коми, що дозволяє виконувати операції з плаваючою комою безпосередньо на апаратному рівні. Це призводить до покращення продуктивності, оскільки операції з плаваючою комою виконуються швидше, ніж у випадку програмного емулювання.

Архітектура armhf підтримує 32-бітну обчислювальну архітектуру ARMv7 та новіші. Вона включає в себе набір інструкцій, що дозволяють працювати з плаваючою комою, такі як додавання, віднімання, множення, ділення та інші. Такі інструкції виконуються на апаратному рівні, що забезпечує швидкість та ефективність операцій з плаваючою комою.

Оскільки armhf використовує апаратну підтримку плаваючої коми, вона є оптимальним вибором для систем, що вимагають інтенсивного використання операцій з плаваючою комою, таких як наукові обчислення, обробка зображень, мультимедіа, графіка та інші додатки, які потребують точності та швидкості при роботі з числами з плаваючою комою.

Архітектура armhf широко використовується у вбудованих системах, мобільних пристроях, планшетах, смартфонах, одноплатних комп'ютерах та інших пристроях на базі ARM. Вона забезпечує оптимальну продуктивність та ефективність при використанні апаратної підтримки плаваючої коми, що дозволяє прискорити роботу з числами з плаваючою комою та покращити виконання додатків, що залежать від цих операцій[10].

4.2 Встановлення та налаштування DHCP серверу

Для встановлення та налаштування DHCP-сервера Dnsmasq на Raspberry Pi потрібно виконати кілька кроків. Ось повний процес:

Встановлення операційної системи: Спочатку вам потрібно встановити операційну систему на Raspberry Pi. Один з найпоширеніших варіантів - це використання Raspbian, офіційної операційної системи Raspberry Pi.

Оновлення системи: Після успішного встановлення операційної системи виконайте оновлення системи за допомогою наступної команди в терміналі:

```
sudo apt-get update sudo apt-get upgrade
```

Це дозволить оновити всі пакети до останньої версії.

Встановлення Dnsmasq: Встановіть пакет Dnsmasq за допомогою наступної команди:

```
sudo apt-get install dnsmasq
```

Це інсталює DHCP-сервер Dnsmasq на ваш Raspberry Pi.

Налаштування Dnsmasq: Після встановлення відредагуйте конфігураційний файл Dnsmasq за допомогою наступної команди:

```
sudo nano /etc/dnsmasq.conf
```

У цьому файлі ви можете налаштувати параметри DHCP, такі як діапазон IP-адрес, маршрутизатор, DNS-сервер та інші. Внесіть потрібні зміни, збережіть файл та вийдіть з редактора.

Перезапуск сервісу: Після налаштування вам потрібно перезапустити сервіс Dnsmasq, щоб внести зміни в силу. Виконайте наступну команду:

```
sudo systemctl restart dnsmasq
```

Перевірка роботи DHCP-сервера: Щоб перевірити, чи працює DHCP-сервер, ви можете підключити інший пристрій до мережі Raspberry Pi і переконатися, що він отримує IP-адресу автоматично.

Це весь процес встановлення та налаштування DHCP-сервера Dnsmasq на Raspberry Pi. За допомогою цього сервера можна автоматично надавати IP-адреси та інші налаштування для підключених пристроїв локальній мережі[6].

4.3 Встановлення hostapd та налаштування конфігурацій для безпровідної точки доступа стандарта IEEE 802.11n/ac

Стандарт IEEE 802.11n/ac є одним з найпоширеніших стандартів безпровідних мереж Wi-Fi, який надає високу швидкість передачі даних та покращену продуктивність порівняно зі старішими стандартами. Основними особливостями стандарту 802.11n/ac є використання множинних антен та технологій з множинним доступом.

Стандарт 802.11n працює на частотах 2,4 ГГц та 5 ГГц і використовує технологію Multiple Input Multiple Output (MIMO), що дозволяє використовувати багато антен для передачі та отримання даних. Це дозволяє досягати вищої швидкості передачі даних та збільшення дальності покриття мережі.

Стандарт 802.11ac є покращеною версією стандарту 802.11n і працює на частоті 5 ГГц. Він використовує технологію MIMO з більшою кількістю антен і впроваджує більш широкі канали передачі даних (80 МГц або 160 МГц), що дозволяє досягати ще вищої швидкості передачі.

Стандарт 802.11n/ac також підтримує більш ефективні методи модуляції, такі як 64-QAM та 256-QAM, що дозволяє передавати більше даних на один символ. Він також використовує технологію Beamforming, що дозволяє зосередити сигнал у напрямку конкретного пристрою, забезпечуючи кращу якість сигналу та швидкість передачі.

Стандарт 802.11n/ac підтримує різні режими роботи, включаючи Access Point (AP) режим, Ad-Hoc режим та режим клієнта, що дозволяє використовувати його як безпровідну точку доступу або підключатися до існуючих безпровідних мереж.

Завдяки високій швидкості передачі даних, покращеній продуктивності та підтримці більшої кількості пристроїв, стандарт 802.11n/ac став популярним в багатьох сферах, включаючи домашні мережі, офіси, громадські місця та інші середовища, де потрібна високоякісна безпровідна

мережа з великою пропускною здатністю.

Для встановлення та налаштування hostapd для безпровідної точки доступу стандарту IEEE 802.11n/ac на Raspberry Pi потрібно виконати кілька кроків. Ось повний процес:

Встановлення hostapd: Виконайте наступну команду в терміналі, щоб встановити пакет hostapd:

```
sudo apt-get install hostapd
```

Це інсталює програму hostapd на ваш Raspberry Pi.

Налаштування мережі: Відредагуйте файл /etc/dhcpcd.conf, щоб налаштувати статичну IP-адресу для вашої безпровідної точки доступу. Додайте наступні рядки в кінець файлу:

interface wlan0 static ip_address=192.168.0.1/24 nohook wpa_supplicant

Це призначить статичну IP-адресу 192.168.0.1/24 для інтерфейсу wlan0 (можете використати іншу IP-адресу, якщо вона вам підходить).

Налаштування hostapd: Створіть новий файл конфігурації для hostapd за допомогою наступної команди:

```
sudo nano /etc/hostapd/hostapd.conf
```

У цьому файлі внесіть наступні налаштування для точки доступу стандарту IEEE 802.11n/ac:

interface=wlan0 driver=nl80211 ssid=YourSSID hw_mode=a channel=36 ieee80211ac=1 wmm_enabled=1 ht_capab=[HT40+][SHORT-GI-20][DSSS_CCK-40]

Замініть "YourSSID" на ім'я вашої безпровідної мережі (SSID). Ви також можете налаштувати канал (channel) та інші параметри згідно вашого вибору.

Налаштування hostapd.conf: Відредагуйте файл /etc/default/hostapd і знайдіть рядок з опцією DAEMON_CONF. Розкоментуйте цей рядок, видаліть коментарний символ "#", і додайте шлях до вашого файлу конфігурації hostapd.conf:

DAEMON_CONF="/etc/hostapd/hostapd.conf"

Налаштування IP-пересування: Включіть IP-пересування за допомогою

наступних команд:

```
sudo nano /etc/sysctl.conf
```

У файлі додайте наступний рядок:

net.ipv4.ip_forward=1

Збережіть зміни і вийдіть з редактора.

Налаштування маскараду: Виконайте наступну команду, щоб налаштувати маскарад (MASQUERADE):

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Це дозволить вашому Raspberry Pi поширювати інтернет-з'єднання через точку доступу.

Збереження налаштувань: Збережіть всі зміни та перезавантажте Raspberry Pi, щоб застосувати налаштування.

Після перезавантаження ваш Raspberry Pi повинен працювати як безпровідна точка доступу стандарту IEEE 802.11n/ac з налаштуваннями, які ви вказали у файлі hostapd.conf. Інші пристрої зможуть підключатися до вашої безпровідної мережі за допомогою вказаного SSID та використовувати його для доступу до мережі та Інтернету.

Будьте уважні при налаштуванні цих параметрів, оскільки неправильні налаштування можуть призвести до проблем з безпровідним з'єднанням або безпекою мережі[5].

4.4 Встановлення та налаштування web серверу lighttpd

Встановлення Lighttpd:

1) Відкрийте термінал або командний рядок на Raspberry Pi.
2) Оновіть список пакетів, виконавши команду: sudo apt update.
3) Встановіть пакет Lighttpd, виконавши команду: sudo apt install lighttpd.
4) Під час встановлення може бути запитано підтвердження. Натисніть "Y" або "Enter", щоб продовжити.

Налаштування Lighttpd:

1) Після успішного встановлення веб-сервера Lighttpd перейдіть до

налаштування.

2) Відкрийте конфігураційний файл Lighttpd за допомогою текстового редактора командою: sudo nano /etc/lighttpd/lighttpd.conf.
3) У цьому файлі ви можете налаштувати різні параметри, такі як кореневий каталог веб-сайту, порт прослуховування, права доступу і багато іншого.
4) Налаштуйте параметри за потребою. Збережіть зміни, натиснувши Ctrl+O, а потім Enter, і вийдіть з редактора, натиснувши Ctrl+X.

Перевірка роботи Lighttpd:

1) Після налаштування переконайтеся, що Lighttpd працює на Raspberry Pi.
2) Відкрийте веб-браузер на комп'ютері або пристрої, підключеному до мережі з Raspberry Pi.
3) Введіть IP-адресу Raspberry Pi або його ім'я хоста в адресному рядку браузера.
4) Якщо все працює належним чином, ви повинні побачити стандартну сторінку привітання Lighttpd.

Розміщення веб-сторінок:

1) Веб-сторінки розміщуються в каталозі /var/www/html/ на Raspberry Pi.
2) Створіть в цьому каталозі файли HTML, CSS, JavaScript і будь-які інші необхідні файли для свого веб-сайту.
3) Переконайтеся, що файли мають правильні права доступу для читання і виконання.

Це базовий процес встановлення та налаштування веб-сервера Lighttpd на Raspberry Pi. Ви можете розширити його можливості, налаштувавши додаткові функції, такі як використання SSL, налаштування віртуальних хостів тощо, залежно від своїх потреб[12].

5 БЕЗПЕКА ЛОКАЛЬНОЇ МЕРЕЖІ

5.1 IDS snort

Для забезпечення безпеки локальної мережі існує кілька важливих аспектів, які необхідно враховувати. Основні з них включають фізичну безпеку, захист від несанкціонованого доступу, захист від вірусів та шкідливих програм, а також забезпечення безпеки даних.

По-перше, фізична безпека локальної мережі передбачає захист фізичних пристроїв, що входять до складу мережі. Це означає, що серверні кімнати, комутатори, маршрутизатори та інші мережеві пристрої повинні бути збережені в безпечному місці з обмеженим доступом. Також важливо фізично захистити кабелі, що використовуються для підключення пристроїв, щоб уникнути несанкціонованого доступу до мережі.

По-друге, захист від несанкціонованого доступу є ключовим аспектом безпеки локальної мережі. Це може бути досягнуто за допомогою механізмів аутентифікації та авторизації. Кожен користувач мережі повинен мати унікальний ідентифікатор та пароль для входу в систему. Крім того, рекомендується використовувати методи шифрування, такі як WPA2 або WPA3, для захисту бездротової мережі Wi-Fi від несанкціонованого доступу.

По-третє, захист від вірусів та шкідливих програм є важливою складовою безпеки локальної мережі. Це можна досягнути за допомогою антивірусного програмного забезпечення, яке буде сканувати всі файли та передачі даних, що входять до мережі, на наявність вірусів та інших шкідливих програм. Оновлення антивірусного програмного забезпечення та оперативна патчінг системи також допоможуть у запобіганні атакам.

По-четверте, забезпечення безпеки даних є критичним аспектом для локальної мережі. Це включає шифрування даних під час їх передачі через мережу, резервне копіювання даних для захисту від втрати та використання методів контролю доступу для обмеження прав доступу до конфіденційної

інформації.

Крім того, важливо використовувати надійне програмне забезпечення та оперативні системи, які постійно оновлюються виробником, щоб усунути виявлені уразливості та забезпечити безпеку мережі.

Загалом, безпека локальної мережі вимагає комплексного підходу та поєднання різних заходів захисту, щоб забезпечити надійну захищеність мережі, пристроїв та даних, що ними обмінюються.

Для рішення данної проблеми було впровадженно IDS snort.

IDS Snort (Intrusion Detection System Snort) є однією з найпопулярніших систем виявлення вторгнень в комп'ютерні мережі. Це відкрите програмне забезпечення, розроблене для виявлення та моніторингу потенційних атак на мережевий трафік.

Snort працює на основі аналізу мережевих пакетів та пошуку сигнатур відомих атак. Він прослуховує мережевий трафік, аналізує його і порівнює з великою базою сигнатур атак, які оновлюються регулярно. Якщо Snort виявляє відповідні сигнатури, він спрацьовує тривогу та сповіщає адміністратора про потенційну загрозу.

Основні можливості IDS Snort включають:

1) Виявлення вторгнень: Snort заснований на сигнатурах, які описують відомі атаки. Він може виявляти широкий спектр атак, включаючи атаки на основі вразливостей, переповнення буфера, отримання несанкціонованого доступу та багато інших.
2) Гнучка настройка: Snort дозволяє адміністратору налаштовувати правила виявлення атак, враховуючи особливості мережі та потенційні загрози. Це дозволяє забезпечити високу точність виявлення та знизити кількість ложних спрацьовувань тривог.
3) Моніторинг мережевого трафіку: Snort дозволяє аналізувати та моніторити мережевий трафік в режимі реального часу. Це дозволяє виявляти потенційні атаки миттєво та приймати необхідні заходи для їх запобігання.

4) Інтеграція з іншими системами: Snort може бути легко інтегрований з іншими системами безпеки, такими як системи реагування на інциденти (IR), системи журналювання подій та системи керування вогнем (Firewall). Це дозволяє створити комплексну систему безпеки мережі.
5) Відкрите програмне забезпечення: Snort є відкритим програмним забезпеченням, що означає, що його вихідний код доступний для перегляду та модифікації. Це надає можливість співтовариству розробників вносити внески, виправляти помилки та вдосконалювати систему в цілому.
6) Режими роботи: IDS Snort пропонує різні режими роботи, включаючи режим виявлення вторгнень (IDS mode) та режим виявлення та запобігання вторгнень (IPS mode). У режимі IDS Snort просто виявляє та сповіщає про потенційні загрози, а в режимі IPS може приймати активні заходи для блокування атак.
7) Підтримка протоколів: IDS Snort підтримує широкий спектр мережевих протоколів, включаючи TCP, UDP, ICMP та інші. Це дозволяє виявляти атаки на різних рівнях мережевого стеку.
8) Розширені можливості: IDS Snort може бути розширений за допомогою різних модулів та додаткових правил. Це дозволяє налаштувати систему під конкретні потреби та врахувати нові загрози та атаки.
9) Аналіз пакетів: IDS Snort забезпечує детальний аналіз мережевих пакетів, включаючи перевірку заголовків, тіл пакетів та іншу інформацію. Це дозволяє виявляти складні атаки, що використовують специфічні вразливості.
10) Спільнота та оновлення: IDS Snort має велику спільноту розробників та користувачів, яка активно співпрацює у покращенні системи. Оновлення сигнатур атак та нових правил випускаються регулярно, щоб забезпечити високу ефективність та актуальність системи.

IDS Snort є потужним інструментом для виявлення вторгнень, який допомагає забезпечити безпеку мережі та захистити її від потенційних загроз. Він є важливою складовою системи безпеки комп'ютерних мереж і дозволяє адміністраторам оперативно реагувати на потенційні атаки та забезпечити безпеку даних. На малюнку ми бачимо роботу IDS snort індитифікація протоколів та шкідливого трафіку.

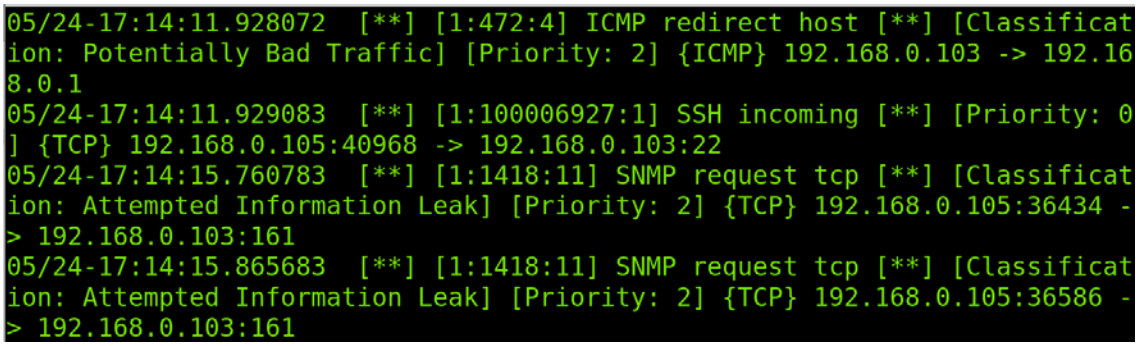


Рисунок 5.1 – IDS snort індитифікація протоколів та шкідливого трафіку.

Snort є одним з найпопулярніших систем виявлення вторгнень (IDS) і володіє кількома перевагами, які роблять його привабливим для багатьох організацій. Відкритий код дозволяє користувачам отримати доступ до вихідного коду, змінювати його за потреби та адаптувати до конкретних вимог мережі, забезпечуючи гнучкість і контроль.

Snort також дозволяє визначати власні правила виявлення вторгнень (rulesets) для виявлення конкретних видів атак або поведінки, що забезпечує гнучкість і можливість налаштування системи під потреби користувача. Він може бути розгорнутий у режимі розподіленої системи, де сенсори Snort розташовуються на різних вузлах мережі для забезпечення виявлення вторгнень на різних рівнях, що особливо корисно для великих мереж з великою кількістю трафіку. Snort також підтримує багато різних протоколів, включаючи IP, TCP, UDP, ICMP і HTTP, що дозволяє аналізувати трафік на різних рівнях і виявляти аномалії або потенційні атаки на різних протоколах. Він

також має підтримку спеціалізованих правил для виявлення конкретних видів атак, таких як DDoS-атаки, SQL-ін'єкції, вторгнення веб-додатків та інші.

Snort також надає можливості реагування на виявлені загрози, де можна налаштувати систему таким чином, щоб вона виконувала різні дії, такі як блокування IP-адреси атакувача або відправку повідомлень адміністратору про виявлену атаку.

Загалом, Snort є потужним і гнучким інструментом для виявлення вторгнень з великим набором можливостей, великою спільнотою користувачів і широкими можливостями налаштування. Однак, вибір найкращої IDS-системи залежить від конкретних потреб і вимог вашої мережі, тому варто оцінити інші альтернативи і порівняти їх з Snort перед прийняттям рішення.

Існує кілька аналогів IDS, які можуть бути розглянуті як альтернативи до Snort. Ось кілька з них:

1) Suricata: Suricata є IDS та IPS (система запобігання вторгненням) з відкритим кодом, яка також базується на правилах. Вона забезпечує високу швидкодію, підтримує багато протоколів і має розширені можливості для аналізу трафіку. Suricata також підтримує багатопоточну обробку, що дозволяє ефективно працювати з великими обсягами даних.
2) Bro (тепер званий Zeek): Bro (Zeek) є мережевим монітором безпеки, який використовує мову Bro для аналізу мережевого трафіку. Він володіє потужними засобами аналізу і дозволяє виявляти вторгнення та аномальну поведінку в мережі. Bro також має велику спільноту користувачів і підтримує розширення за допомогою плагінів.
3) OSSEC: OSSEC є системою виявлення вторгнень і управління безпекою з відкритим кодом, яка пропонує розширені функції моніторингу логів, інтеграцію з системами журналювання та реагування на виявлені загрози. Вона підтримує розподілені

архітектури та має можливості для аналізу в реальному часі.

Визначення того, який IDS краще для мого проекту, залежить від конкретних потреб і контексту використання. Однак, є деякі фактори, які можуть робити Snort бажанним вибором порівняно з Suricata, Zeek та OSSEC для деяких організацій:

1) Досвід та популярність: Snort є одним з найпопулярніших і визнаних систем виявлення вторгнень. Його широко використовують і він має значну спільноту користувачів і розробників. Це означає, що ви зможете знайти багато матеріалів, документації, плагінів та підтримку з боку спільноти.
2) Широкі можливості налаштування: Snort надає гнучкість і можливості налаштування системи під конкретні потреби. Ви можете визначати власні правила виявлення вторгнень і адаптувати систему до вашого середовища. Це дозволяє забезпечити точне виявлення і аналіз атак та аномалій.
3) Розширюваність та інтеграція: Snort має можливості для розширення за допомогою плагінів та інтеграції з іншими системами безпеки. Ви можете розширити функціональність Snort, додавши додаткові модулі або інтегруючи його з іншими інструментами, що полегшує управління безпекою в комплексних середовищах.
4) Мережевий розподіл: Snort підтримує розподілені системи, де сенсори можуть бути розташовані на різних вузлах мережі. Це дає можливість виявляти вторгнення на різних рівнях мережі і забезпечувати захист від широкого спектру загроз.

Однак, важливо враховувати, що Suricata, Zeek та OSSEC також мають свої переваги і можуть бути кращими виборами для деяких сценаріїв. Наприклад, Suricata володіє високою швидкодією та розширеними можливостями для аналізу трафіку, Zeek спеціалізується на мережевому моніторингу, а OSSEC має додаткові функції управління безпекою. Тому

перед вибором IDS варто ретельно оцінити свої потреби та провести порівняльний аналіз функціональності та можливостей кожного рішення.

IDS Snort ідентифікує атаки та інциденти за допомогою правил виявлення вторгнень, які встановлюються адміністратором або користувачем. Ці правила визначають певні шаблони або патерни, що характеризують певні види атак або аномальну поведінку в мережі. Коли Snort аналізує мережевий трафік, він порівнює цей трафік з встановленими правилами для виявлення відповідних атак або аномалій.

Snort підтримує два основних типи правил:

1) Правила засновані на сигнатурах: Ці правила використовують конкретні сигнатури атак або зловмисних дій, які вже відомі та документовані. Коли Snort знаходить збіг між сигнатурою, вказаною в правилі, і пакетом даних, він визначає це як потенційну атаку.
2) Правила засновані на аномаліях: Ці правила визначають нормальну поведінку мережі та її користувачів. Вони аналізують мережевий трафік і шукають відхилення або аномальні патерни, які можуть вказувати на потенційну атаку або порушення безпеки.

Після виявлення атаки або інциденту, Snort може спрацювати реакційно і прийняти відповідні заходи. Це може включати блокування IP-адреси атакувача, запис журналу подій, надсилання сповіщень адміністратору або інші дії, встановлені адміністратором системи[11].

5.2 MITM атаки та боротьба з ними

MITM або "Man-in-the-Middle" (людина посередині) атака - це вид атаки, в якому зловмисник встановлює позицію посередника між двома комунікуючими сторонами в мережі з метою перехоплення, зміни або підробки комунікації між ними. В результаті цієї атаки зловмисник може отримати доступ до конфіденційної інформації, такої як логіни, паролі, фінансові дані, персональні повідомлення тощо.

У MITM атаки зловмисник може використовувати різні методи, включаючи наступні:

1) ARP спуфінг: Зловмисник відправляє фальшиві ARP-пакети, щоб перехопити мережевий трафік, спрямований до і від цільових пристроїв.
2) DNS отруєння: Зловмисник зламує систему DNS і перенаправляє запити на фальшиві DNS-сервери, що дозволяє йому контролювати та перехоплювати мережевий трафік.
3) Використання публічних Wi-Fi точок доступу: Зловмисник створює фальшиву Wi-Fi точку доступу, яка надається під ім'ям знайомої мережі, щоб перехоплювати мережевий трафік користувачів.
4) SSL/TLS злам: Зловмисник може використовувати атаки на протоколи шифрування, такі як SSL/TLS, для розшифрування та перехоплення захищеного трафіку.
5) Фішингові атаки: Зловмисник може надіслати фальшиві електронні листи або створити підроблені веб-сторінки для отримання конфіденційних даних від користувачів.

Наслідки MITM атак можуть бути серйозними, оскільки зловмисник може мати доступ до чутливої інформації, зламати аутентифікацію та авторизацію, виконувати шкідливі дії в мережі та імітувати легітимні комунікації.

Крадіжку конфіденційної інформації: Зловмисник, який перехоплює комунікацію між двома сторонами, може отримати доступ до конфіденційних даних, таких як логіни, паролі, фінансові дані, персональна інформація і т.д. Ця інформація може використовуватися для злочинних дій, таких як крадіжка особистих аккаунтів, шахрайство або ідентифікаційна крадіжка.

Зміну комунікації: Зловмисник може модифікувати передані дані між сторонами. Це може призвести до недостовірної інформації, зміни угод або контрактів, спотворення команд інтернет-банкінгу або переказів коштів. Змінена комунікація може також вплинути на безпеку системи або призвести

до вразливостей.

Підробку ідентичності: Зловмисник може використовувати MITM атаку, щоб підробити ідентичність одного з комунікуючих сторін. Це може дозволити зловмиснику набути неправомірний доступ до ресурсів або привілеїв, до яких він не має права. Зловмисник може використовувати підроблену ідентичність для виконання шкідливих дій від імені легітимної сторони.

MITM атакак націлена на моніпуляцю протокола ARP. ARP (Address Resolution Protocol) - це протокол на мережевому рівні, що використовується в комп'ютерних мережах для вирішення проблеми відповідності між IP-адресами і MAC-адресами. В основі роботи ARP лежить завдання встановлення відповідності між логічними адресами IP та фізичними адресами MAC.

Коли пристрій в мережі намагається надіслати пакет до певної IP-адреси, він спочатку перевіряє своє кешування ARP для визначення відповідного MAC-адреси. Якщо запису немає в кеші ARP, пристрій відправляє ARP-запит, що містить запит до відповідної IP-адреси із запитом про повернення відповідного MAC-адреси.

ARP-запит поширюється по всій мережі, і пристрій з відповідною IP-адресою відповідає на нього ARP-відповіддю, в якій вказує свою MAC-адресу. При отриманні ARP-відповіді, пристрій оновлює своє кешування ARP, додаючи новий запис, який містить IP-адресу та відповідну MAC-адресу.

ARP грає важливу роль в мережевому взаємодії, дозволяючи пристроям в мережі знаходити інші пристрої та встановлювати з ними з'єднання. Протокол дозволяє налаштовувати таблиці ARP на пристроях, що мають можливість кешування ARP-записів, що полегшує процес визначення фізичної адреси пристрою з відомою IP-адресою. ARP є невід'ємною частиною функціонування TCP/IP мереж і використовується у різних мережевих середовищах, включаючи Ethernet та Wi-Fi.

Хоча ARP виконує важливі функції для забезпечення комунікації в мережах, він також може бути використаний для здійснення атак на безпеку, зокрема MITM (Man-in-the-Middle) атак. Під час MITM атаки зловмисник перехоплює ARP-запити і ARP-відповіді, використовуючи їх для отримання контролю над мережевим трафіком і пересилання його без попередження пристроям-учасникам мережі.

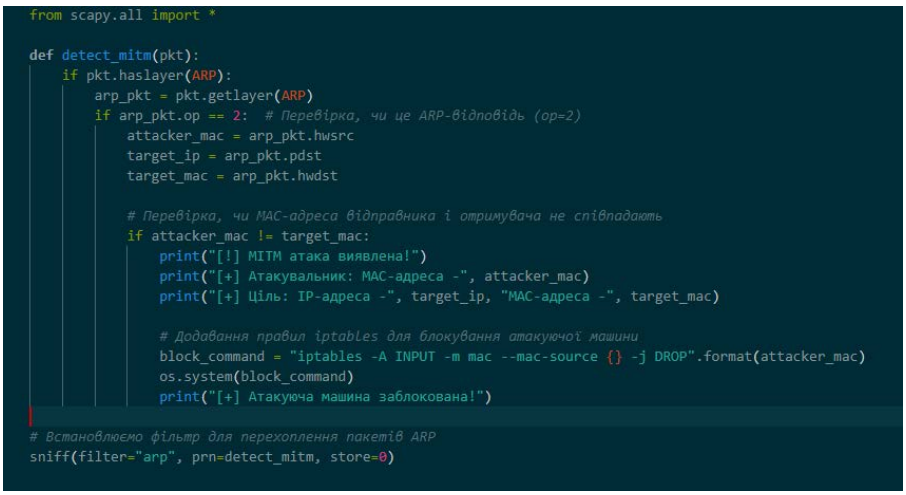
Вплив на репутацію і бізнес: MITM атаки можуть негативно вплинути на репутацію компаній, організацій або індивідуальних користувачів. Якщо конфіденційні дані або приватна інформація стають доступними, це може призвести до втрати довіри клієнтів, витоку комерційних та конкурентних даних і негативного впливу на ділову діяльність.

Пошкодження інфраструктури: MITM атаки можуть негативно вплинути на інфраструктуру мережі або системи, в яких вони відбуваються. Зловмисник може спотворити або перенаправити мережевий трафік, що призводить до порушення нормального функціонування системи або зниження продуктивності.

Загроза національній безпеці: MITM атаки можуть бути використані зловмисниками або злочинними групами для здійснення шпигунства, зламу критичних інфраструктур або проведення кібератак з метою національної шкоди.

Усі ці наслідки роблять MITM атаки серйозною загрозою для безпеки та конфіденційності комунікаційних систем і вимагають вжиття ефективних заходів захисту та превентивних заходів проти таких атак.

Ось приклад скрипта, який детектує атаку MITM і додає правила до iptables для блокування атакуючої машини. Зверніть увагу, що виконання цих дій може вимагати привілеїв адміністратора.



```python
from scapy.all import *

def detect_mitm(pkt):
    if pkt.haslayer(ARP):
        arp_pkt = pkt.getlayer(ARP)
        if arp_pkt.op == 2:  # Перевірка, чи це ARP-відповідь (op=2)
            attacker_mac = arp_pkt.hwsrc
            target_ip = arp_pkt.pdst
            target_mac = arp_pkt.hwdst

            # Перевірка, чи MAC-адреса відправника і отримувача не співпадають
            if attacker_mac != target_mac:
                print("[!] MITM атака виявлена!")
                print("[+] Атакувальник: MAC-адреса -", attacker_mac)
                print("[+] Ціль: IP-адреса -", target_ip, "MAC-адреса -", target_mac)

                # Додавання правил iptables для блокування атакуючої машини
                block_command = "iptables -A INPUT -m mac --mac-source {} -j DROP".format(attacker_mac)
                os.system(block_command)
                print("[+] Атакуюча машина заблокована!")

# Встановлюємо фільтр для перехоплення пакетів ARP
sniff(filter="arp", prn=detect_mitm, store=0)
```

Рисунок 5.2 – Реалізація скрипту на Python для блокування MITM атак.

Цей скрипт використовує бібліотеку Scapy для перехоплення та аналізу пакетів ARP, як і у попередньому скрипті. Після виявлення MITM атаки, він виводить повідомлення та виконує команду iptables для додавання правила, яке блокує атакуючу машину за її MAC-адресою. Переконайтеся, що ви виконуєте цей скрипт з правами адміністратора, оскільки використовується команда iptables[4].

6 ДОСЛІДЖЕННЯ АНАЛОГІВ HOT-SPOT

На ринку існує безліч роутерів, які мають підтримку IPS (системи запобігання вторгненням). Деякі з них виготовляються різними виробниками, такими як TP-Link, D-Link і 3Com. Ці компанії відомі своєю довголітньою історією і надійністю своїх продуктів.

6.1 Аналіз аналогів роутерів з підтримкою IPS

TP-Link пропонує широкий асортимент роутерів, включаючи моделі, такі як Archer C5400X, Archer C3150 V2 і Archer AX6000, які мають вбудовану підтримку IPS. Ці моделі є потужними і забезпечують надійний захист вашої мережі.

D-Link також відомий своїми продуктами з підтримкою IPS. Моделі, такі як DIR-895L/R, DIR-882 і DIR-878, володіють високою продуктивністю та широким функціоналом. Вони забезпечують ефективний захист мережі від потенційних загроз і вторгнень.

3Com - ще один виробник роутерів, який пропонує моделі з підтримкою IPS. Наприклад, 3Com OfficeConnect VPN Firewall та 3Com SuperStack 3 Firewall є надійними рішеннями для захисту мережі від вторгнень. Вони мають функціонал, який дозволяє виявляти та блокувати потенційно небезпечний трафік.

6.2 Порівняння цінового діапазону обладнання

TP-LINK Archer C5400X

- Технологія бездротового з'єднання: Wi-Fi 5 (802.11ac)
- Швидкість передачі даних: до 5400 Мбіт/с
- Кількість антен: 8 внутрішніх антен
- Порти: 4 LAN-порти Gigabit Ethernet, 1 WAN-порт Gigabit Ethernet
- Функції безпеки: вбудована система IPS, фаервол, контроль доступу,

VPN-підтримка

- Інші особливості: процесор з 1.8 ГГц, підтримка MU-MIMO, Beamforming
- Ціна на ринку: 21849 грн

D-LINK DIR-895L/R

- Технологія бездротового з'єднання: Wi-Fi 5 (802.11ac)
- Швидкість передачі даних: до 5300 Мбіт/с
- Кількість антен: 8 зовнішніх антен
- Порти: 4 LAN-порти Gigabit Ethernet, 1 WAN-порт Gigabit Ethernet
- Функції безпеки: вбудована система IPS, фаервол, контроль доступу, VPN-підтримка
- Інші особливості: двоядерний процесор, підтримка MU-MIMO, Beamforming, USB-порти
- Ціна: 11444 грн

3Com OfficeConnect VPN Firewall

- Швидкість передачі даних: до 1000 Мбіт/с
- Кількість антен: внутрішні антени
- Порти: 4 LAN-порти Fast Ethernet, 1 WAN-порт Fast Ethernet
- Функції безпеки: вбудована система IPS, фаервол, контроль доступу, VPN-підтримка
- Інші особливості: підтримка VLAN, QoS, SPI
- Ціна: 10526 грн

RASPBERRY PI 4B 2GB:

Процесор: 64-бітний чотирьохядерний процесор ARM Cortex-A72 з тактовою частотою 1,5 ГГц.

- Пам'ять: 2 ГБ оперативної пам'яті LPDDR4.
- З'єднання: два порти USB 3.0, два порти USB 2.0, два порти micro-HDMI для підключення до двох моніторів з роздільною здатністю до 4K, порт Gigabit Ethernet, порт microSD для зберігання операційної системи та даних.

- Бездротові можливості: підтримка Wi-Fi 802.11ac і Bluetooth 5.0.
- Графіка: вбудований графічний процесор VideoCore VI з підтримкою OpenGL ES 3.x і 4K відтворенням відео.
- Сховище: можливість підключення зовнішнього сховища через порт USB або використання карти microSD.
- Операційна система: можливість встановлення різних операційних систем, таких як Raspbian, Ubuntu, а також інших дистрибутивів Linux.
- GPIO: 40 GPIO-портів для підключення різних датчиків, пристроїв та розширень.
- Живлення: підтримка живлення через USB-C адаптер з напругою 5 В і струмом 3 А.
- Ціна: 1300 грн.

Як бачимо, ціни на ці роутери завищені, а твердження виробника про те, що їх роутери забезпечать зростання швидкості та безпеки, не зовсім вірні, особливо щодо безпеки. Часто їх системи IPS ґрунтуються лише на кількох правилах iptables і не мають динамічної, простої та гнучкої системи, як у моєму проекті.

Для створення свого проекту я витратив 1300 грн на придбання міні-комп'ютера Raspberry Pi 4B 2GB і реалізував на ньому точку доступу hot-spot з гнучкою та динамічною системою IDS. Такий підхід дозволяє мені забезпечити більшу ефективність та контроль над безпекою мережі. Крім того, використання Raspberry Pi дозволяє мені налаштовувати його під свої потреби і розширювати його можливості за допомогою різних додаткових модулів і компонентів. Такий саморобний підхід не тільки економить кошти, але і дозволяє мені мати повний контроль над своєю мережею та забезпечити потрібний рівень безпеки.

7 WEB ІНТЕРФЕЙС

Для мого проекту був створений простий та зручний веб-інтерфейс, що дозволяє контролювати процеси точки доступу hot-spot. Цей інтерфейс розроблений з використанням мов програмування, таких як PHP та Python, мови розмітки HTML та мови стилів CSS. Завдяки цьому, користувач може легко налаштовувати і керувати параметрами точки доступу, використовуючи простий та зрозумілий інтерфейс веб-сторінки. Це забезпечує зручну та ефективну роботу з проектом та спрощує взаємодію з точкою доступу hot-spot.

На малюнку ми можемо побачити роботу WEB інтерфейсу а саме взаємодія з сервесами hot-spot а саме IDS , Dnsmasq, Hostapd.

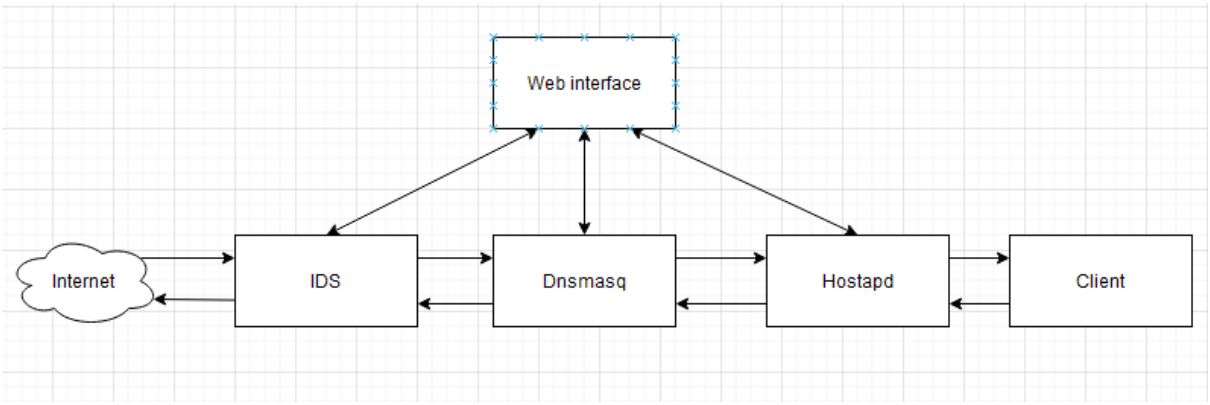


Рисунок 7.1 – IR модель WEB інтерфейсу

Крім того, веб-інтерфейс підтримує різні функціональні можливості, такі як налаштування безпеки, керування доступом до мережі, моніторинг підключених пристроїв і статистика використання ресурсів. Користувач може легко переглядати активних користувачів, блокувати небажаних або незареєстрованих пристроїв, а також налаштовувати фільтри для контролю вмісту. Також, інтерфейс забезпечує можливість встановлення гнучких правил та політик безпеки, що дозволяє забезпечити високий рівень захисту мережі.

Загалом, завдяки простому та функціональному веб-інтерфейсу, який

підтримується мовами програмування PHP та Python, мовою розмітки HTML та мовою стилів CSS, мій проект забезпечує зручне керування та контроль над точкою доступу hot-spot. Користувачі зможуть налаштовувати та керувати різними параметрами, забезпечуючи безпеку та ефективну роботу мережі.

Після авторизації веб-інтерфейсу, ми зустрічаємо ліву панель керування, де маємо наступні розділи:

1) Головна панель (Dashboard): Тут ми можемо отримати огляд загальної інформації та стану системи.
2) Hotspot: В цьому розділі ми можемо налаштувати параметри точки доступу Hotspot, включаючи аутентифікацію, шифрування, керування користувачами та інші налаштування.
3) DHCP Server: Тут ми можемо керувати DHCP-сервером і налаштовувати параметри автоматичного призначення IP-адрес для підключених пристроїв.
4) Ad Blocking: У цьому розділі можна активувати та налаштувати блокування реклами для забезпечення кращої веб-перегляду.
5) Networking: Тут ми можемо налаштовувати мережеві параметри, такі як WAN-підключення, DNS-сервери, перенаправлення портів та інші мережеві налаштування.
6) WIFI client: В цьому розділі ми можемо налаштувати підключення до існуючої бездротової мережі як клієнт.
7) OpenVPN: Тут можна налаштувати підключення до віртуальної приватної мережі (VPN) за допомогою протоколу OpenVPN.
8) WireGuard: В цьому розділі можна налаштувати підключення до мережі за допомогою протоколу WireGuard VPN.
9) Аутентифікація (Authentication): Тут можна налаштувати методи аутентифікації для доступу до системи.
10) Використання даних (Data usage): В цьому розділі можна переглядати статистику використання даних та керувати обмеженнями

швидкості.

11) Система (System): Тут можна налаштовувати загальні параметри системи, включаючи часовий пояс, мову, оновлення програмного забезпечення та інші налаштування.
12) Про AirLink (About AirLink): В цьому розділі надається інформація про систему AirLink, версію програмного забезпечення та інші деталі.

Завдяки цим розділам та їх функціональності, ми можемо зручно налаштовувати та керувати різними аспектами нашої мережі за допомогою web-інтерфейсу.

7.1 Головна панель - Dashboard

На головній панелі (Dashboard) ми отримуємо огляд загальної інформації та стану системи. Тут можна переглянути загальний обсяг трафіку, виміряний у мегабайтах, який був використаний за певний період часу. Крім того, у розділі WIFI client відображається статус підключення до інших точок доступу, рівень сигналу та швидкість передачі даних.

У розділі Connection Device можна переглянути список клієнтів, які підключені до нашої точки доступу. Тут відображаються ім'я хоста (ім'я клієнта), IP-адреса в локальній мережі та фізична адреса клієнта (MAC-адреса). Ця інформація дозволяє нам зручно контролювати та керувати підключеними пристроями, переглядати їх стан та статистику, що сприяє ефективному управлінню мережею.

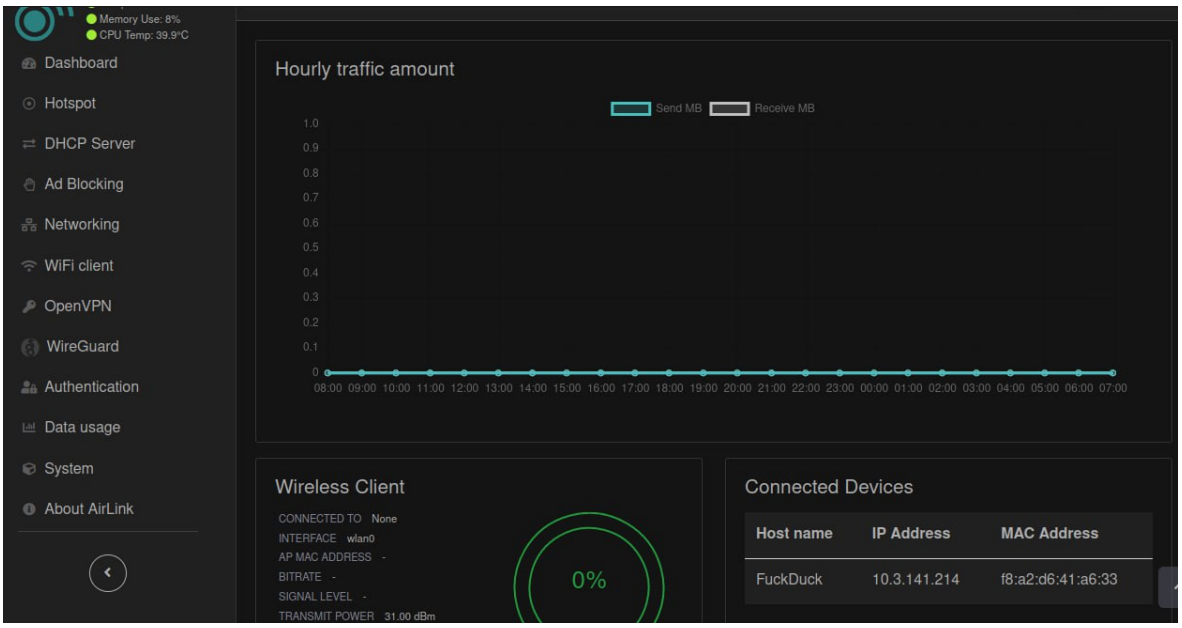


Рисунок 7.2 – Головний екран WEB інтерфейсу

7.2 Налаштування точки доступу у розділі Hotspot

При натисканні на розділ Hotspot ми зможемо налаштувати нашу точку доступу. Тут ми можемо вибрати інтерфейс, через який буде здійснюватися з'єднання з клієнтом, встановити назву точки доступу (SSID) та вибрати робочу частоту, на якій точка доступу буде працювати.

Ці налаштування дозволяють нам зручно контролювати та налаштовувати параметри нашої точки доступу, забезпечуючи стабільну та надійну роботу мережі.

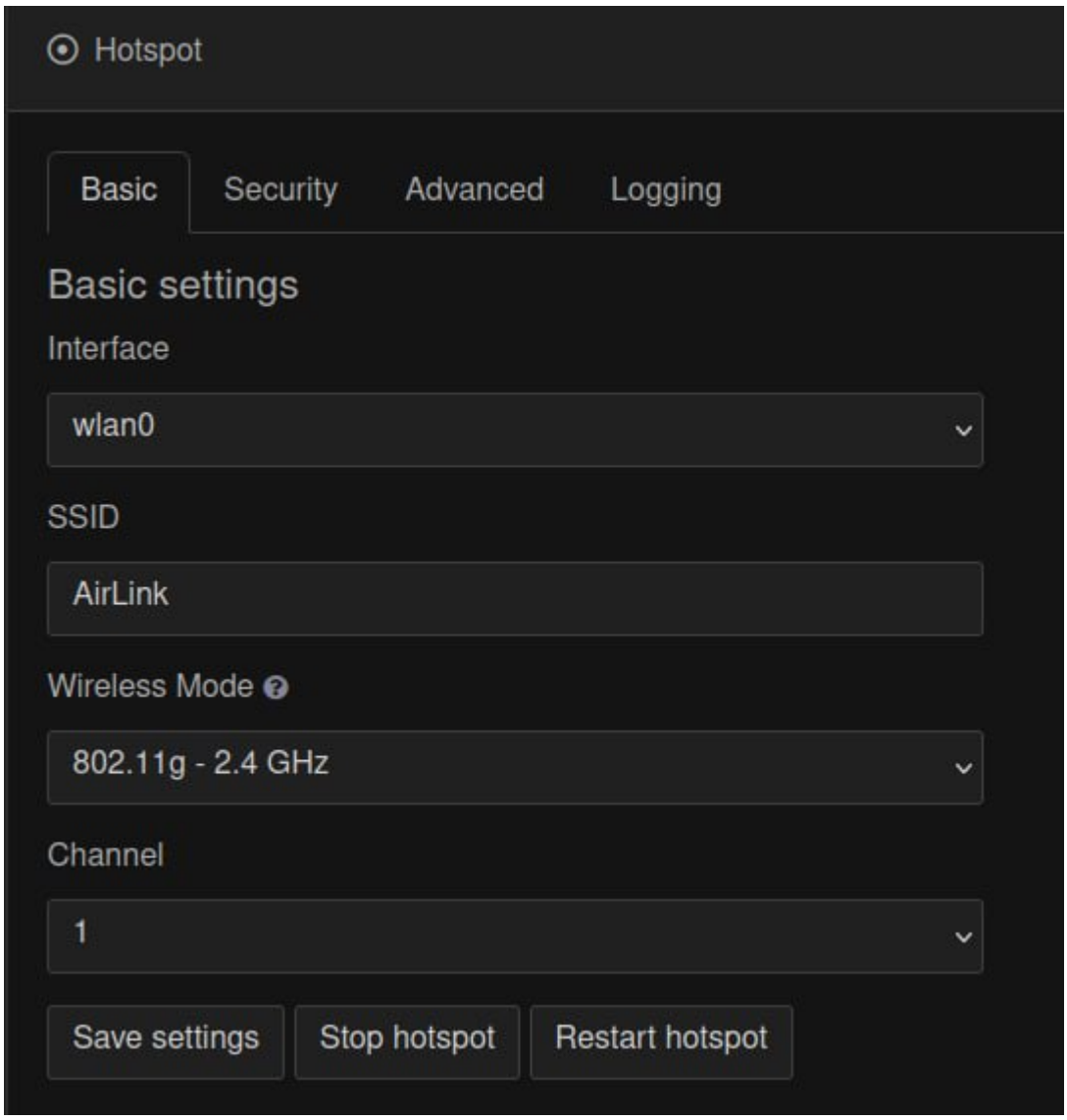


Рисунок 7.3 – Стандартне налаштування Hotspot

У розділі Security ми можемо вибрати протокол безпеки, тип шифрування та встановити свій пароль (PSK) для нашої точки доступу. Крім того, в цьому розділі генерується QR-код, за допомогою якого ми можемо зручно підключитися до нашої точки доступу без необхідності вводити пароль.

Ці налаштування забезпечують високий рівень безпеки мережі і зручність використання, дозволяючи легко і швидко підключатися до точки доступу за допомогою QR-коду.

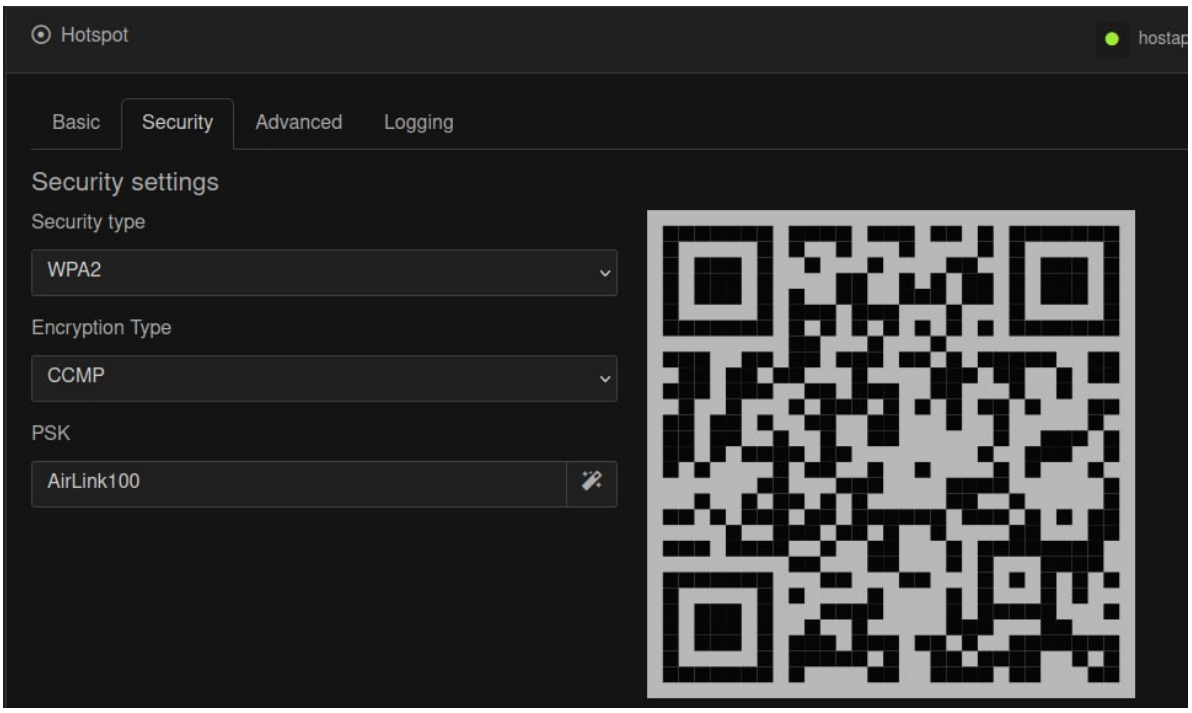


Рисунок 7.4 – Налаштування безпеки hot-spot

У розділі Advanced ми можемо встановити додаткові налаштування для точки доступу, такі як:

1) Bridge AP mode: режим мосту точки доступу.
2) WiFi client AP mode: режим точки доступу як WiFi-клієнта.
3) Hide SSID in broadcast: приховати назву точки доступу при трансляції.
4) Beacon interval (стандартне значення 100): інтервал передачі маячків.
5) Disable low ack: вимкнути низькі підтвердження передачі.
6) Transmit power: потужність передачі.
7) Maximum number of clients: максимальна кількість клієнтів.
8) Country code: код країни.

Також у розділі доступні кнопки Save Setting (зберегти налаштування), Stop Hotspot (зупинити точку доступу) та Restart Hotspot (перезапустити точку доступу).

Ці налаштування дозволяють нам докладніше настроїти режим роботи точки доступу залежно від наших потреб та забезпечити оптимальну продуктивність мережі.

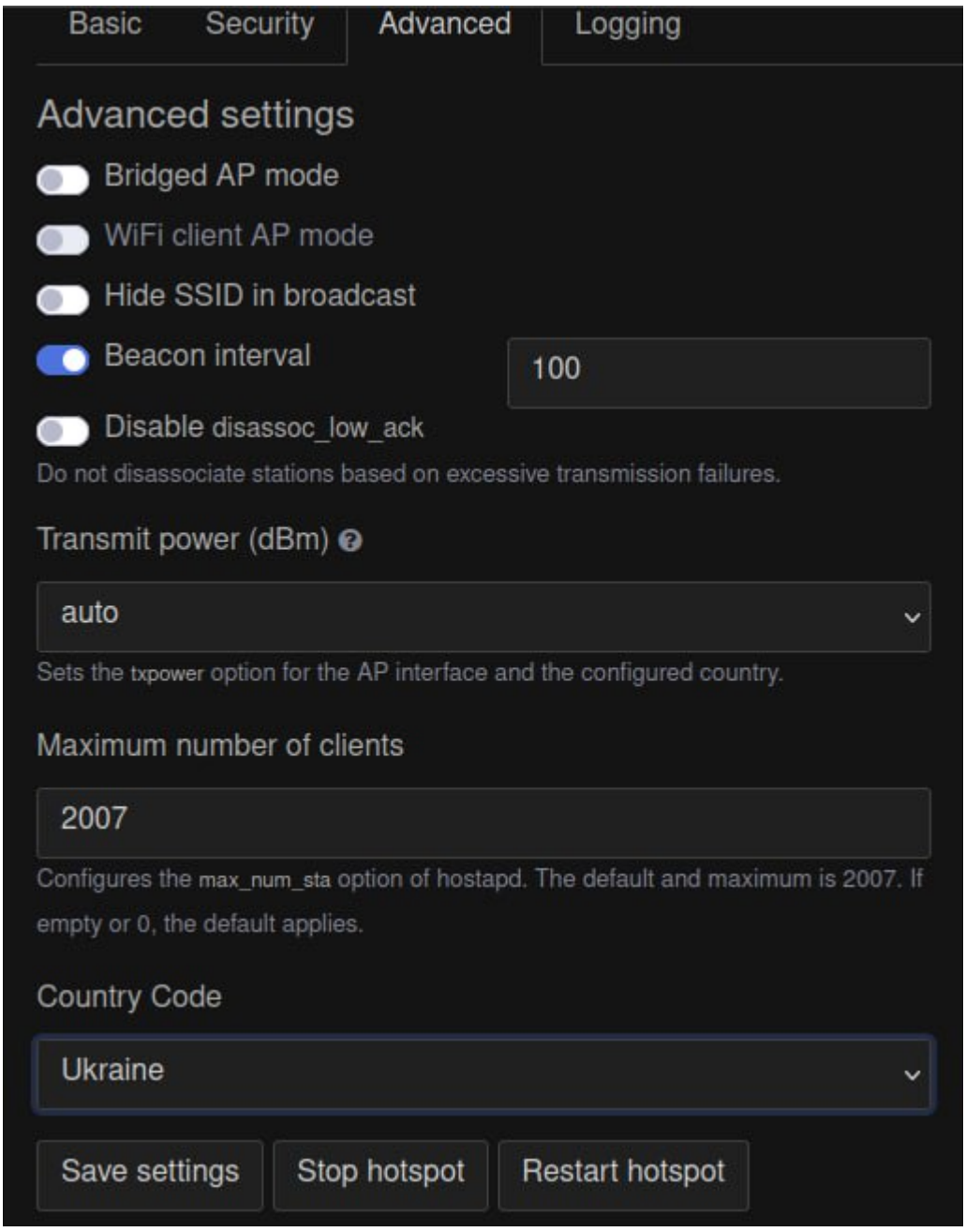


Рисунок 7.5 – Розширене налаштування hot-spot

7.3 Налаштування DHCP Server

У розділі DHCP Server ми можемо налаштувати сервер для автоматичного призначення IP-адрес комп'ютерам і пристроям, що підключаються до мережі. Тут ми можемо вибрати інтерфейс, через який буде здійснюватися взаємодія, обрати тип призначення IP-адрес (статичний або динамічний) і налаштувати мережеві параметри, такі як діапазон IP-адрес і DNS-сервер.

Ці налаштування дозволяють нам зручно керувати процесом призначення IP-адрес і забезпечити належну роботу мережі, забезпечуючи належні налаштування IP-адрес та DNS-сервера для підключених пристроїв.

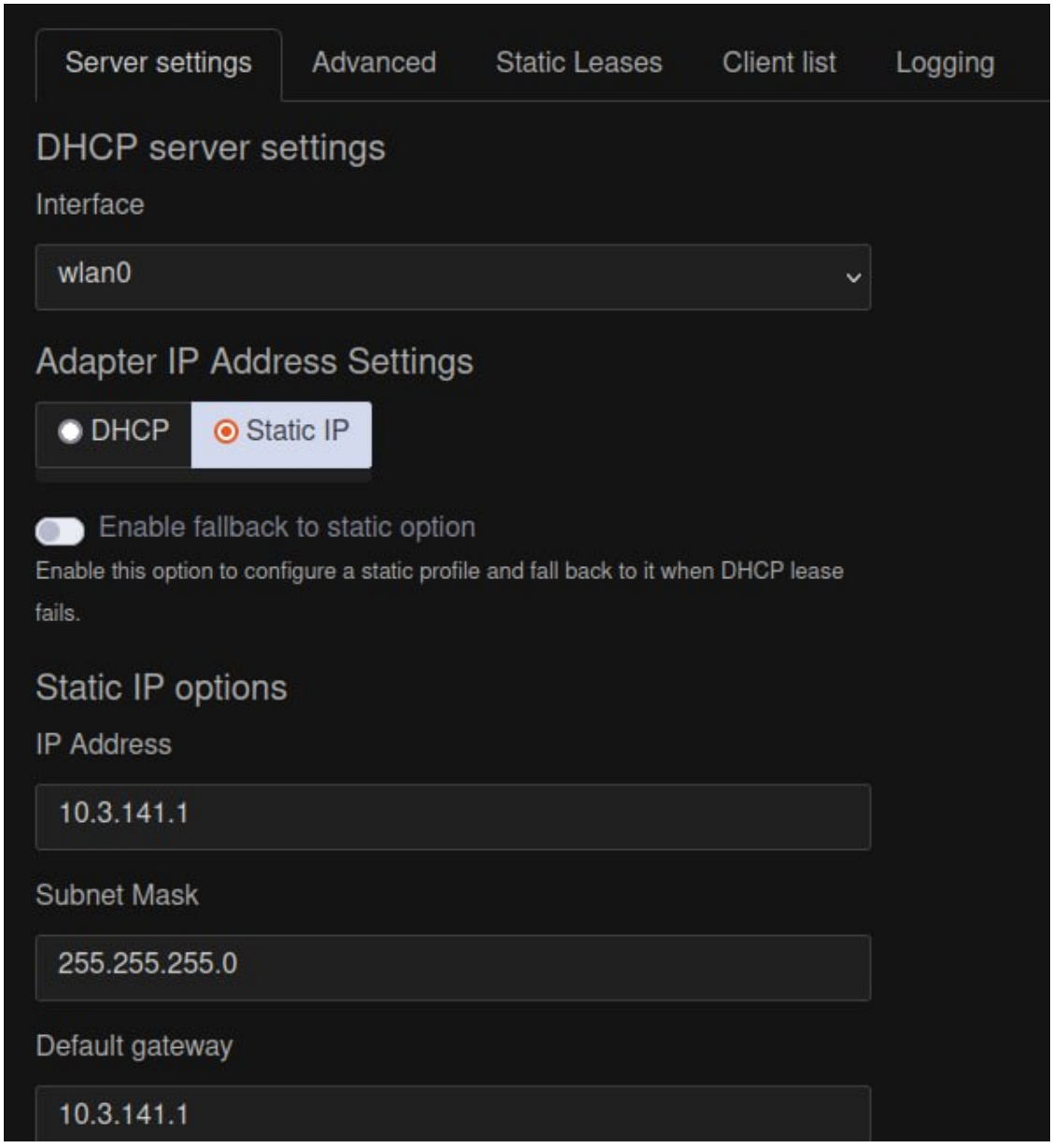


Рисунок 7.6 – Налаштування Dnsmasq

У розділі Advanced є можливість налаштувати DNS-сервер, зокрема:

- Only ever query DNS servers configured below: можливість встановити, що будуть використовуватися тільки DNS-сервери, які налаштовані нижче.
- Add upstream DNS server: можливість додати зовнішній DNS-сервер

для використання в якості додаткового джерела запитів.

Ці налаштування дозволяють точно налаштувати поведінку DNS-сервера та забезпечити використання конкретних серверів для запитів DNS.

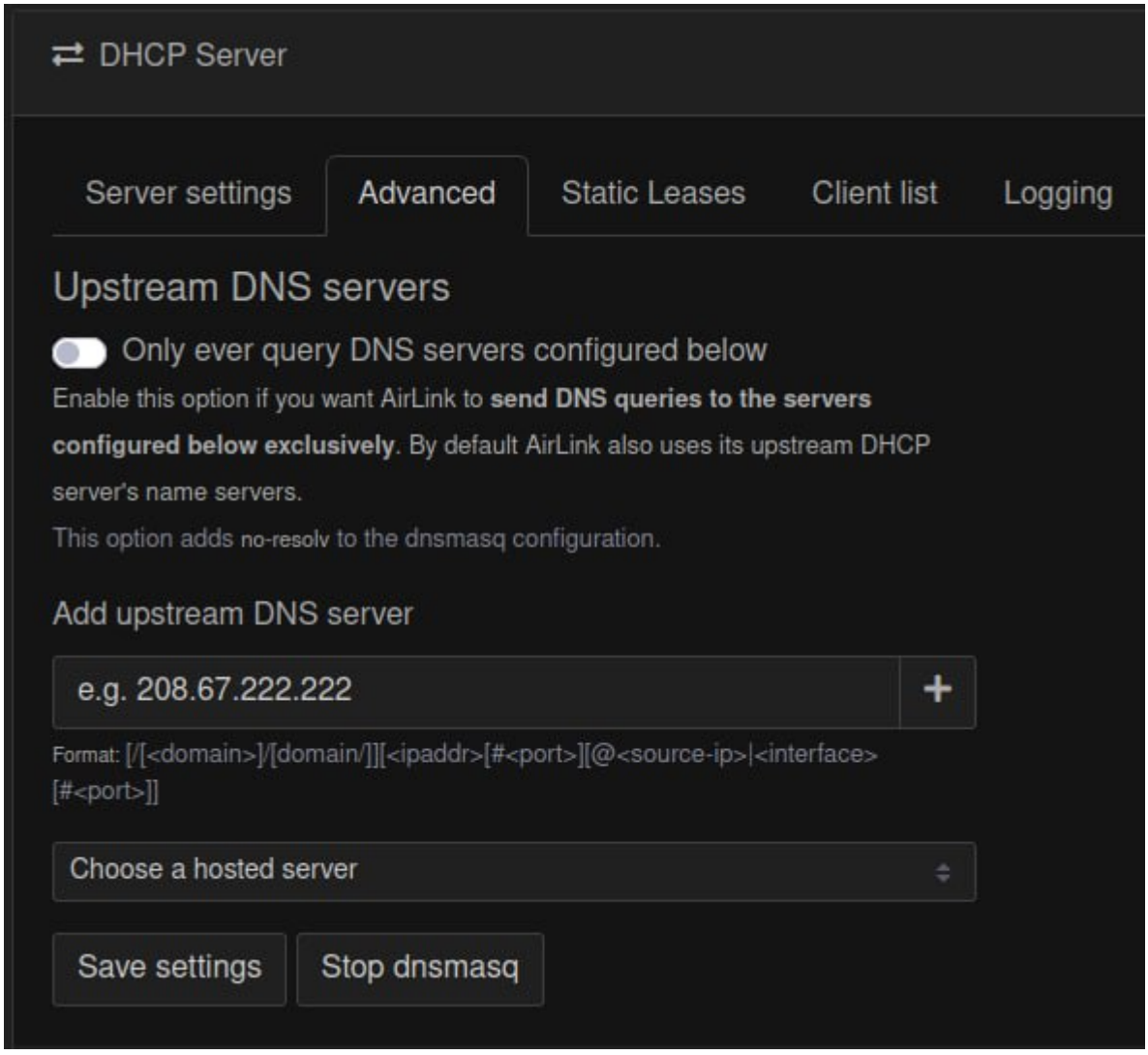


Рисунок 7.7 – Розширене налаштування Dnsmasq

У розділі Static Leases ми маємо можливість призначити певному клієнту статичну IP-адресу, яка буде закріплена за ним. Це означає, що незалежно від того, як часто клієнт підключається до мережі, йому буде надано той самий IP-адрес. Це корисна функція, особливо в тих випадках, коли нам потрібно мати стабільне ідентифікаційне значення для певного пристрою в мережі.

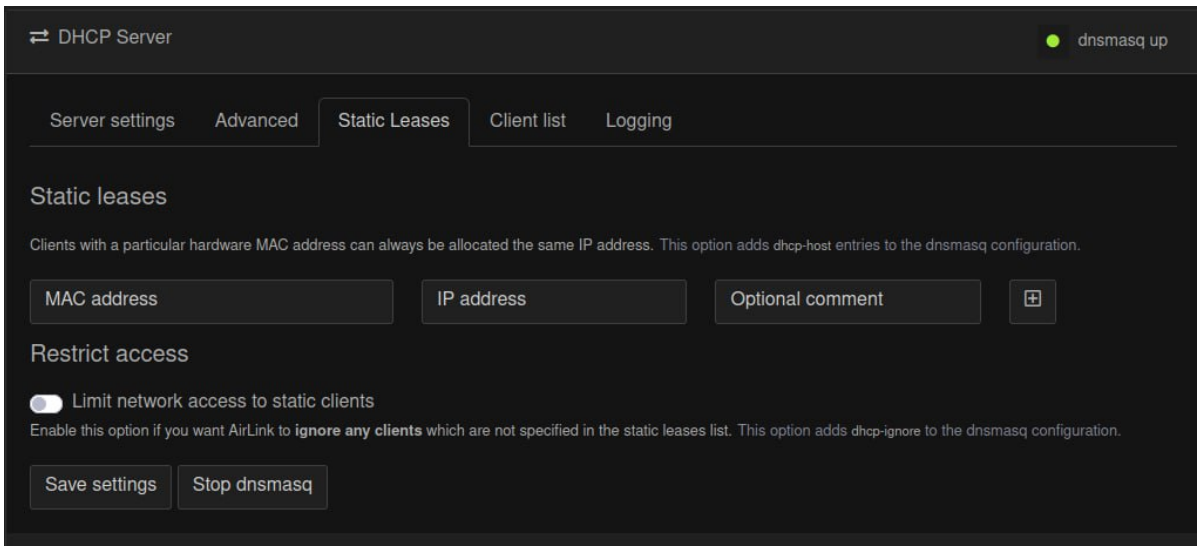


Рисунок 7.8 – Налаштування резервованої IP адреси

У розділі Client List ми можемо переглянути список клієнтів, які підключилися до нашої точки доступу. Інформація, яку ми можемо побачити, включає MAC-адресу, IP-адресу, ім'я хоста та ідентифікатор клієнта. Це дозволяє нам зручно відстежувати та ідентифікувати пристрої, які використовують нашу мережу, і встановлювати контроль над ними.

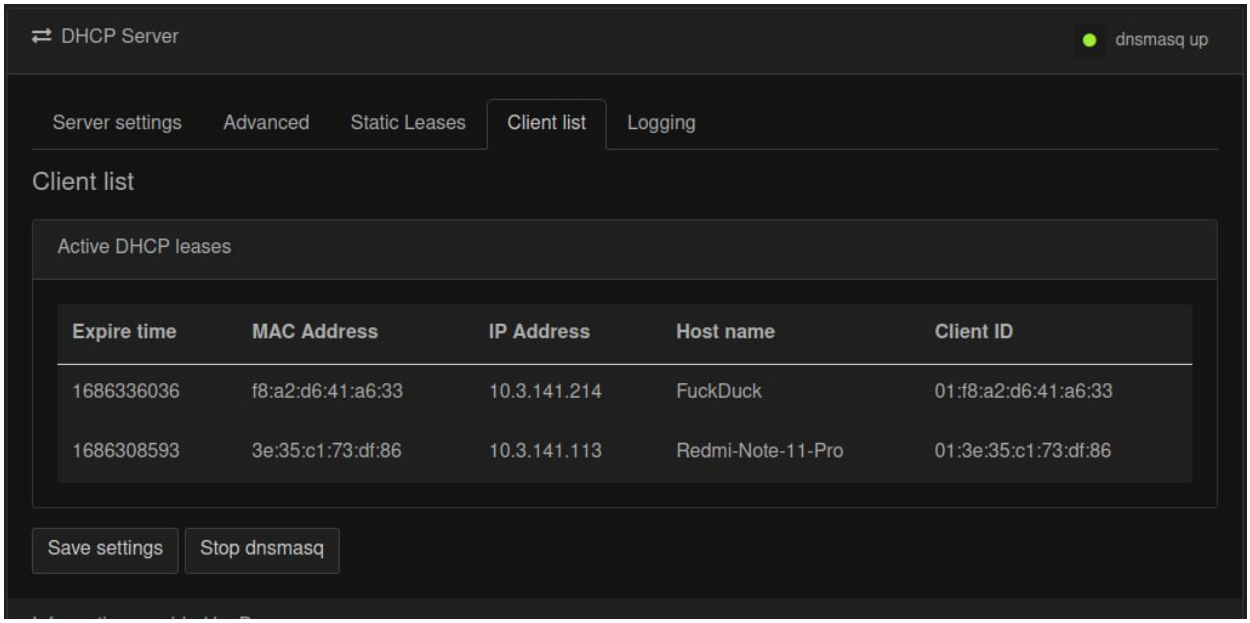


Рисунок 7.9 – Список клієнтів

У розділі Logging ми маємо можливість переглянути логи DHCP- та

DNS-серверів. Це дозволяє нам зберігати записи подій, пов'язаних з розподілом IP-адрес DHCP-сервером та вирішенням DNS-запитів DNS-сервером. Логи надають важливу інформацію про здійснені дії та дозволяють нам відстежувати події, пов'язані з мережевими процесами, що відбуваються на роутері. Це може бути корисно для виявлення проблем, аналізу мережевого трафіку та налагодження мережевих налаштувань.

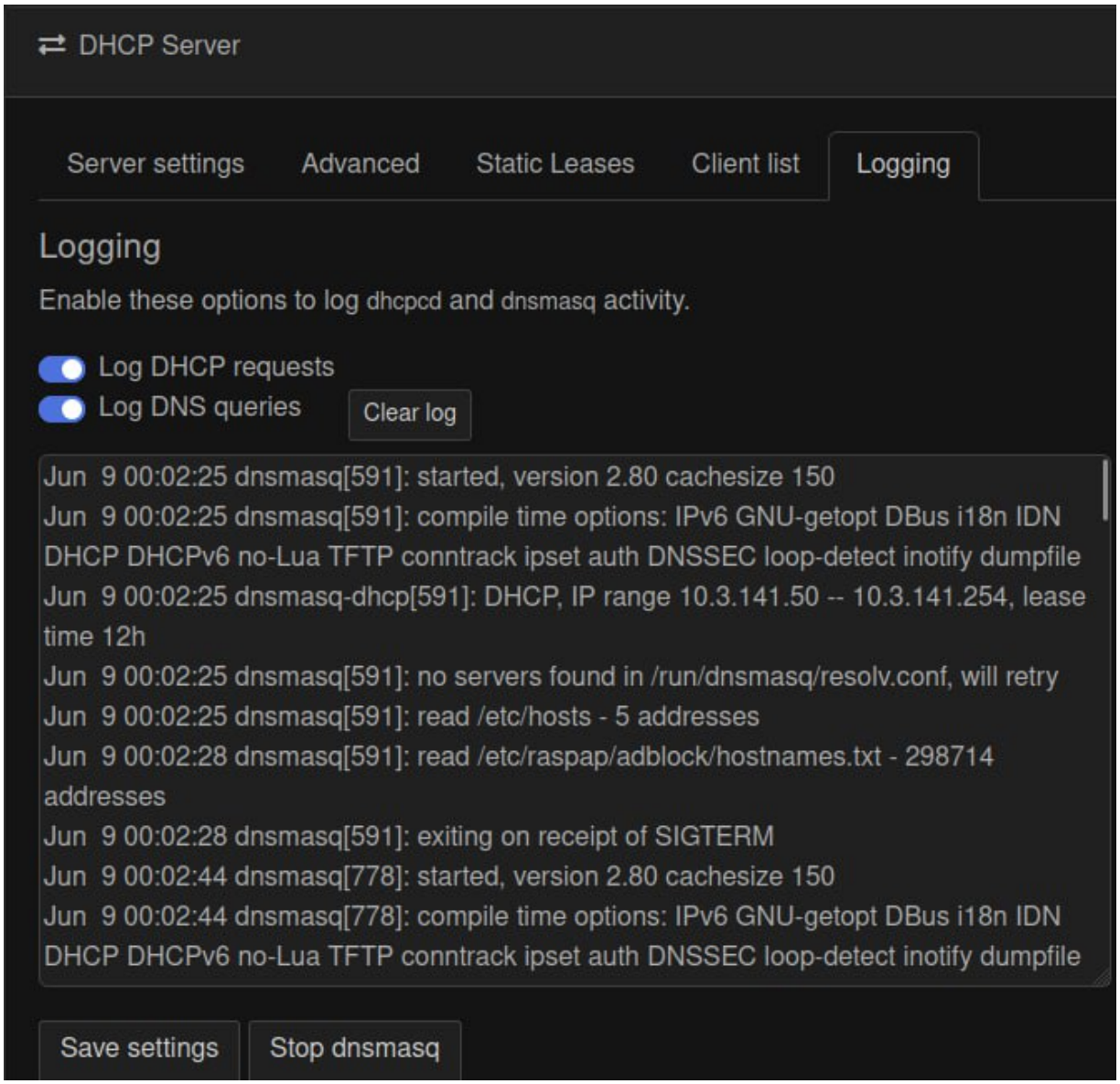


Рисунок 7.10 – Журнал подій Dnsmasq

7.4 Блокування реклами

У розділі Ad Blocking ми маємо можливість налаштувати блокування реклами. Ми можемо активувати функцію блокування реклами (Enable Blocklist) для ефективного фільтрування небажаної реклами на пристроях, підключених до нашої мережі. Це дозволяє нам поліпшити відживлення в Інтернеті, зменшити витрати трафіку і забезпечити більш приємний та безпечний веб-досвід для користувачів. Можна також налаштувати власний список блокування для вибору конкретних джерел реклами, які ми хочемо блокувати.

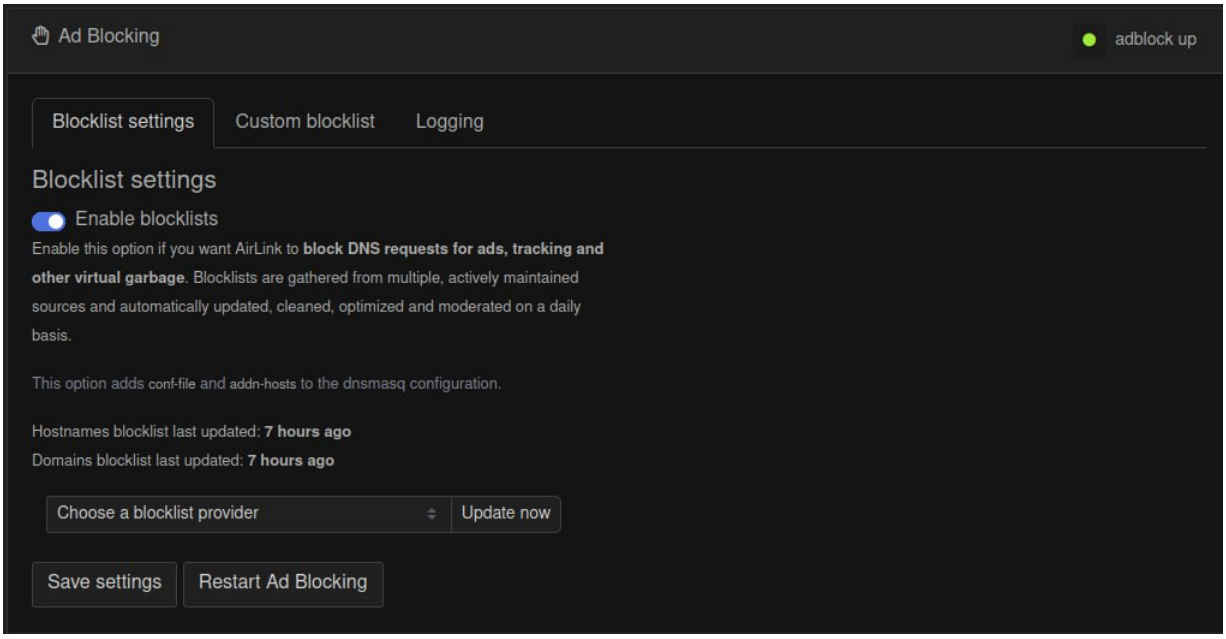


Рисунок 7.11 – Блокування реклами.

У розділі Custom Blocklist ми маємо можливість налаштувати власний список блокування для ефективного блокування рекламних ресурсів. Ми можемо додати IP-адреси або доменні імена в список, які бажаємо блокувати. Це дозволяє нам контролювати і блокувати специфічні сайти або сервери, пов'язані з небажаною рекламою або небезпечними джерелами. Налаштування власного блок-списку дозволяє нам персоналізувати рівень блокування і забезпечити більш точне фільтрування відповідно до наших

вимог та потреб.

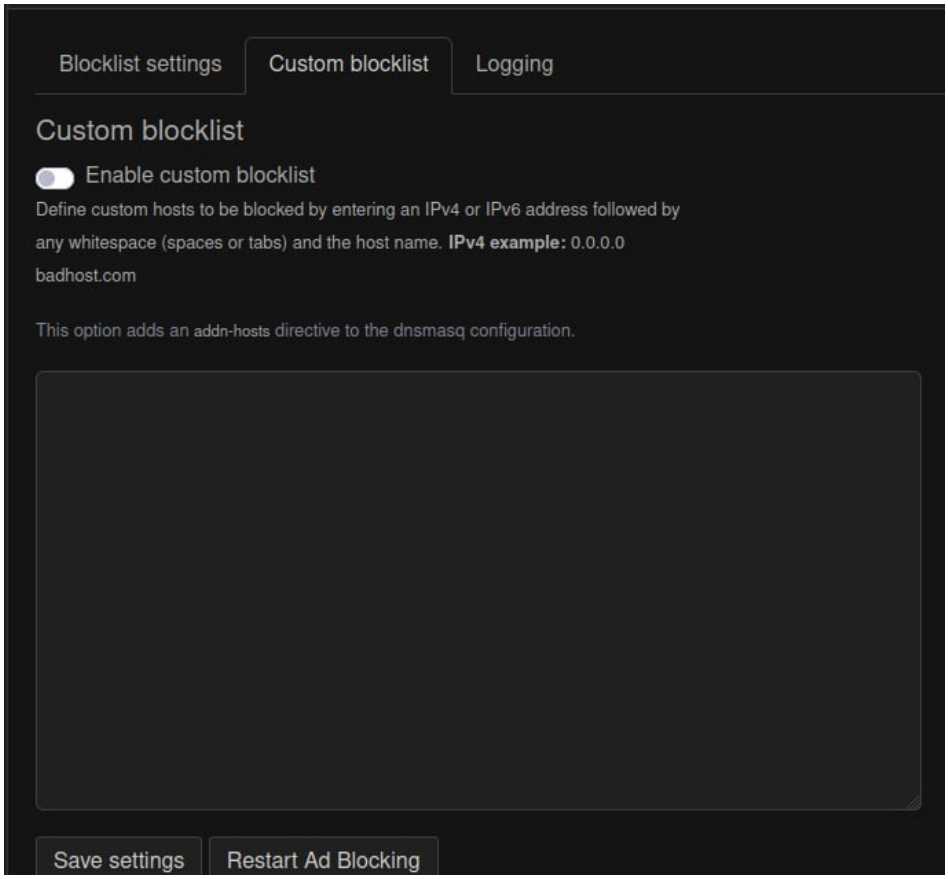


Рисунок 7.12 – Налаштування свого списка для блокування реклами

У розділі Logging ми маємо можливість переглянути інформацію про блокування реклами. Тут ми знаходимо журнал, який відображає події, пов'язані з блокуванням рекламних ресурсів. Ми можемо переглянути записи про рекламні запити, які були заблоковані, і отримати детальну інформацію про джерело, тип або домен реклами, яка була заблокована. Це дозволяє нам вести контроль і перевіряти ефективність нашого блокування реклами, а також ідентифікувати потенційно небезпечні джерела рекламних матеріалів.

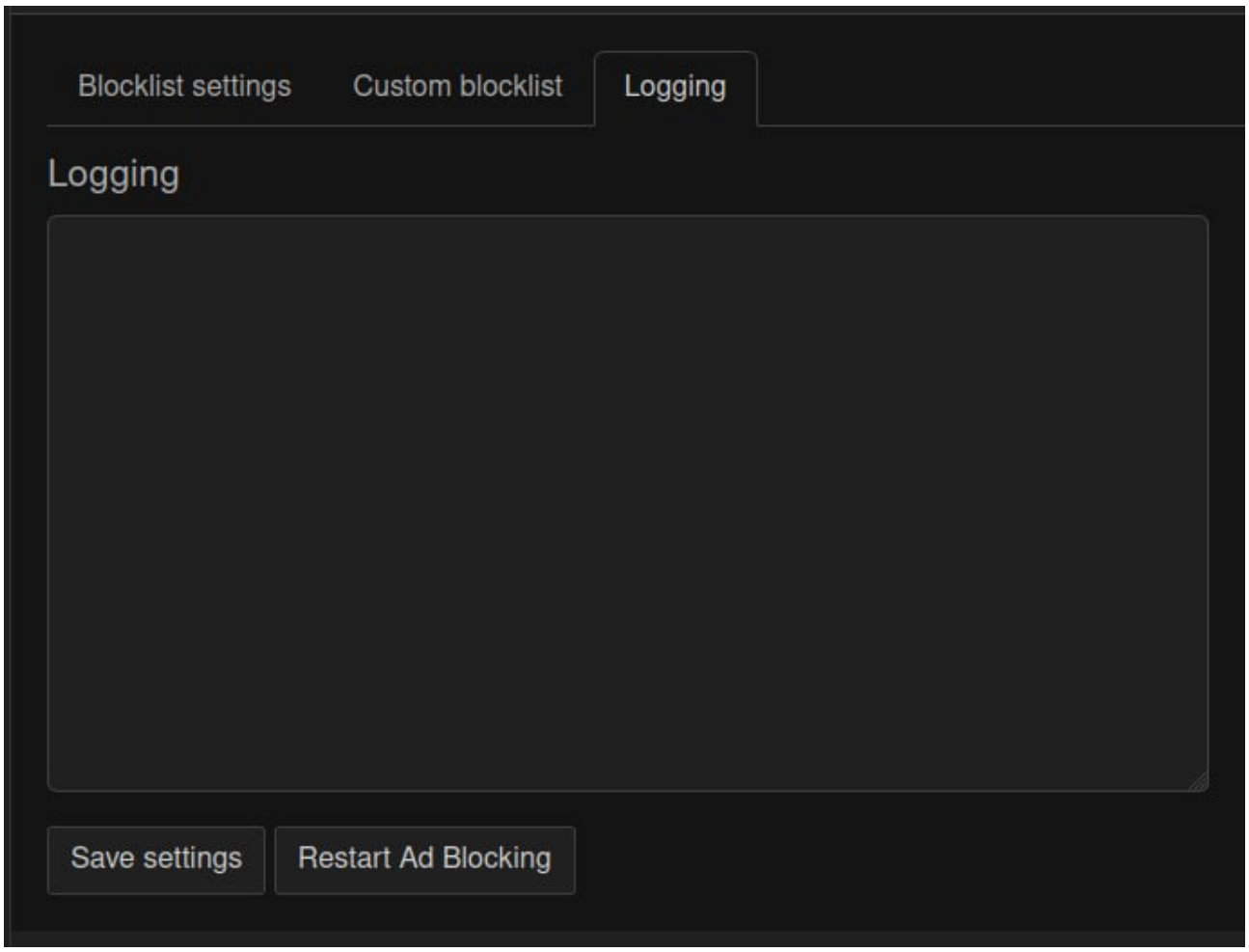


Рисунок 7.13 – Журнал подій блокування реклами.

7.5 Мережеві інтерфейси

У розділі Networking ми можемо переглянути статус нашого інтернет-з'єднання, а також отримати інформацію про мережеві інтерфейси. Тут ми зможемо перевірити, чи підключено наше пристрій до Інтернету, переглянути швидкість передачі даних та стан з'єднання. Крім того, ми матимемо можливість отримати детальну інформацію про IP-адресу, MAC-адресу та інтерфейси, які використовуються для мережевого підключення. Це дозволяє нам краще розуміти нашу мережеву конфігурацію та виконувати необхідні налаштування.

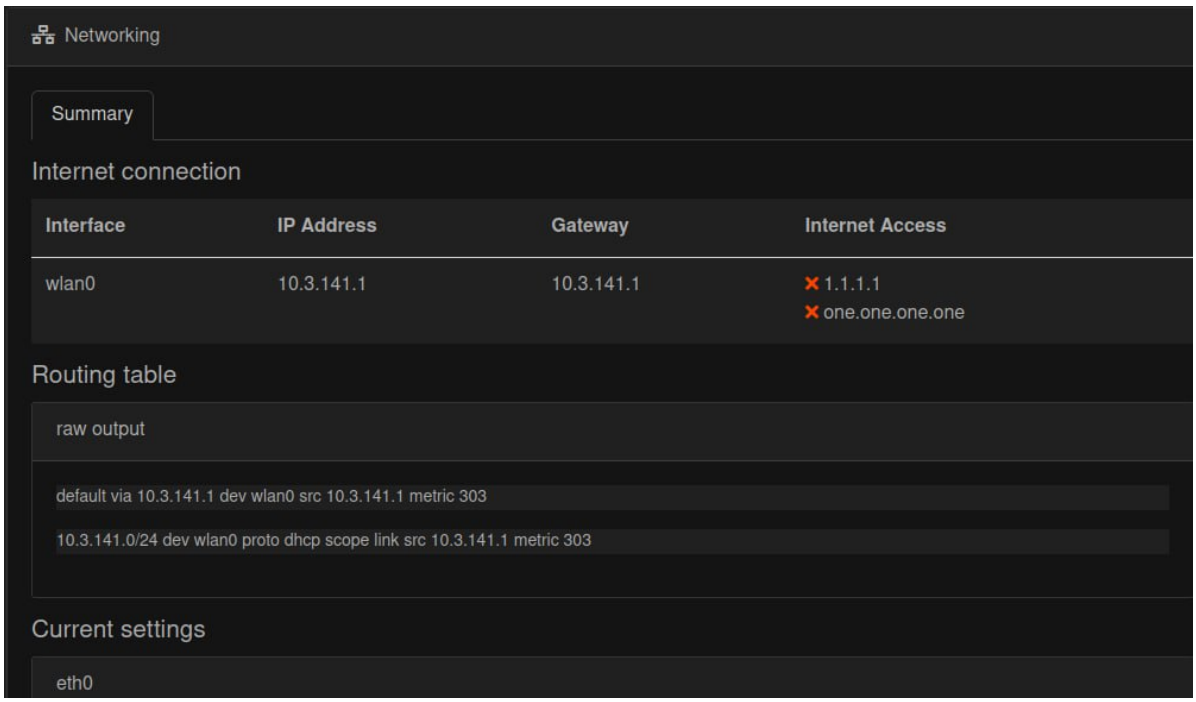


Рисунок 7.14 – Інформація про мережеві інтерфейси

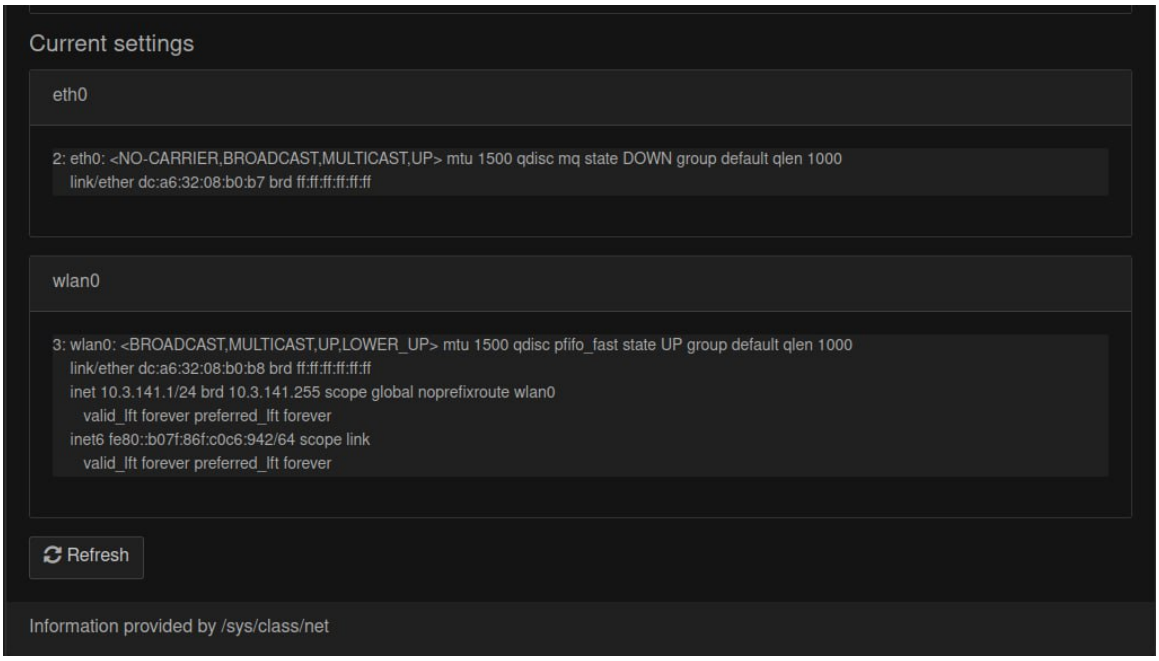


Рисунок 7.14.1 – Інформація о мережевих інтерфейсів

7.5 OpenVPN

У розділі OpenVPN ми можемо налаштувати з'єднання з VPN-сервером. Тут ми маємо можливість ввести свої облікові дані - логін та пароль, або використовувати сертифікат для автентифікації. OpenVPN надає нам

можливість безпечного і зашифрованого з'єднання з віддаленим сервером, що дозволяє нам зберігати приватність та безпеку під час передачі даних. Налаштування OpenVPN дозволяє нам встановлювати параметри з'єднання, такі як серверну адресу, порт та шифрування, що забезпечує гнучкість і контроль над нашими VPN-з'єднаннями.

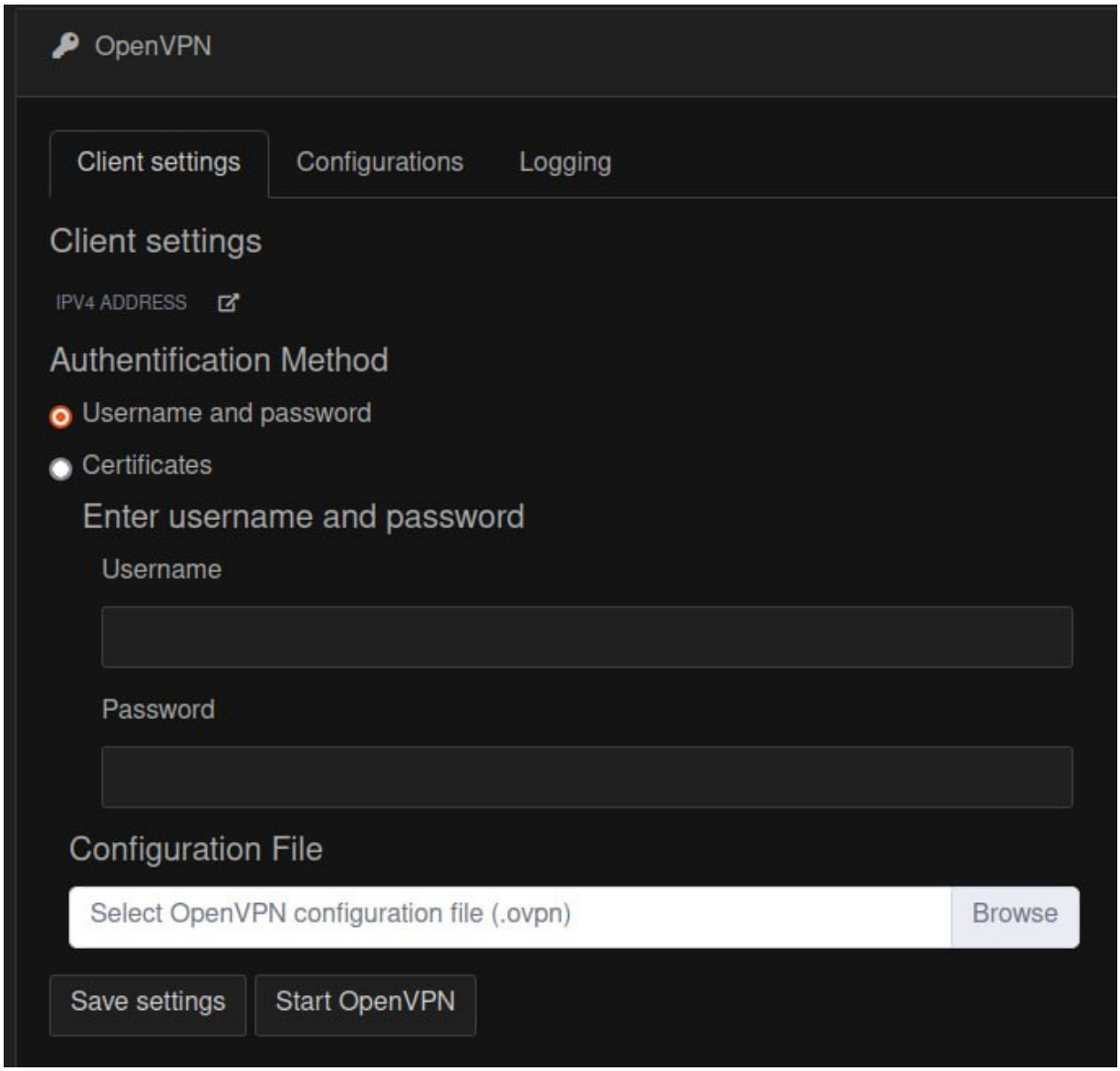


Рисунок 7.15 – Налаштування OpenVPN

У розділі Configuration ми маємо можливість включити або відключити службу OpenVPN. Це дозволяє нам контролювати стан сервісу і активувати або зупиняти його за потреби. З використанням цього налаштування ми

можемо включати OpenVPN, коли ми хочемо встановити з'єднання з віддаленим сервером за допомогою VPN, або вимикати його, коли нам не потрібне з'єднання. Ця функція надає нам гнучкість і зручність у керуванні режимом роботи OpenVPN на нашому пристрої.

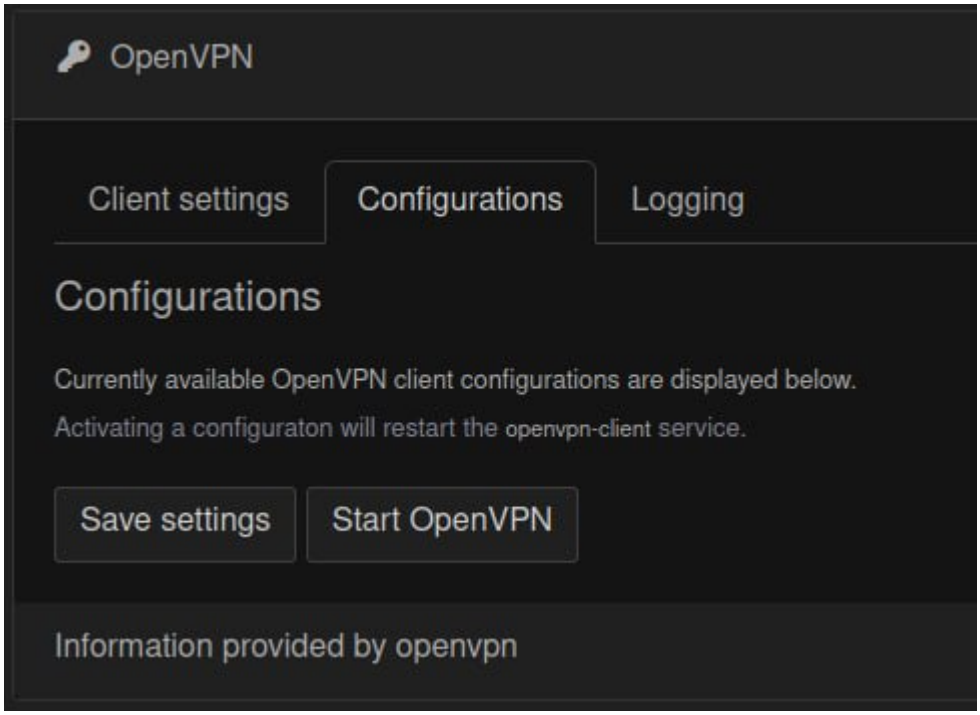


Рисунок 7.16 – Налаштування кофігурації OpenVPN

У розділі Logging ми зможемо переглянути логи сервісу OpenVPN. Це дозволяє нам отримати доступ до докладної інформації про події, що відбуваються в сервісі OpenVPN. За допомогою цієї функції ми можемо перевірити статус та роботу нашого VPN-з'єднання, виявити можливі проблеми та відстежувати активність з'єднання. Це надає нам засоби для моніторингу та аналізу даних, що допомагають забезпечити стабільну та безпечну роботу нашого VPN-з'єднання.

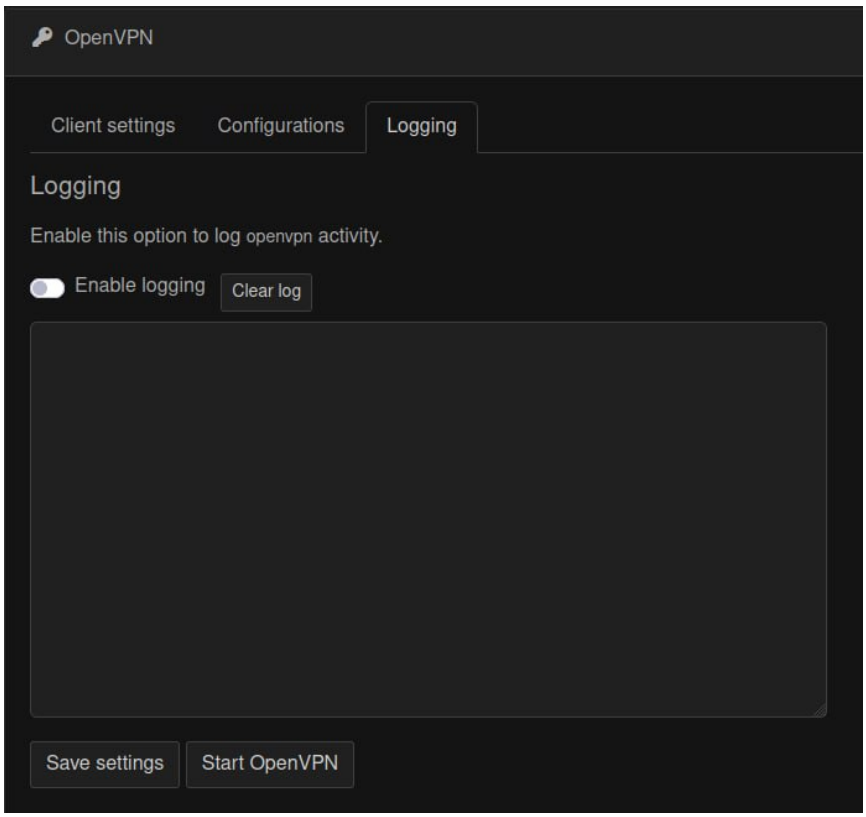


Рисунок 7.17 – Журнал подій OpenVPN

7.6 WireGuard

У розділі WireGuard ми маємо можливість налаштувати та підключитись до VPN. Налаштування проводиться шляхом використання конфігураційного файлу або шляхом введення логіну та паролю. WireGuard надає простий і зручний спосіб налаштування та забезпечує безпечне з'єднання з VPN-сервером. За допомогою цієї функції ми можемо отримати доступ до захищеної мережі та забезпечити конфіденційність та безпеку наших даних при передачі через Інтернет.

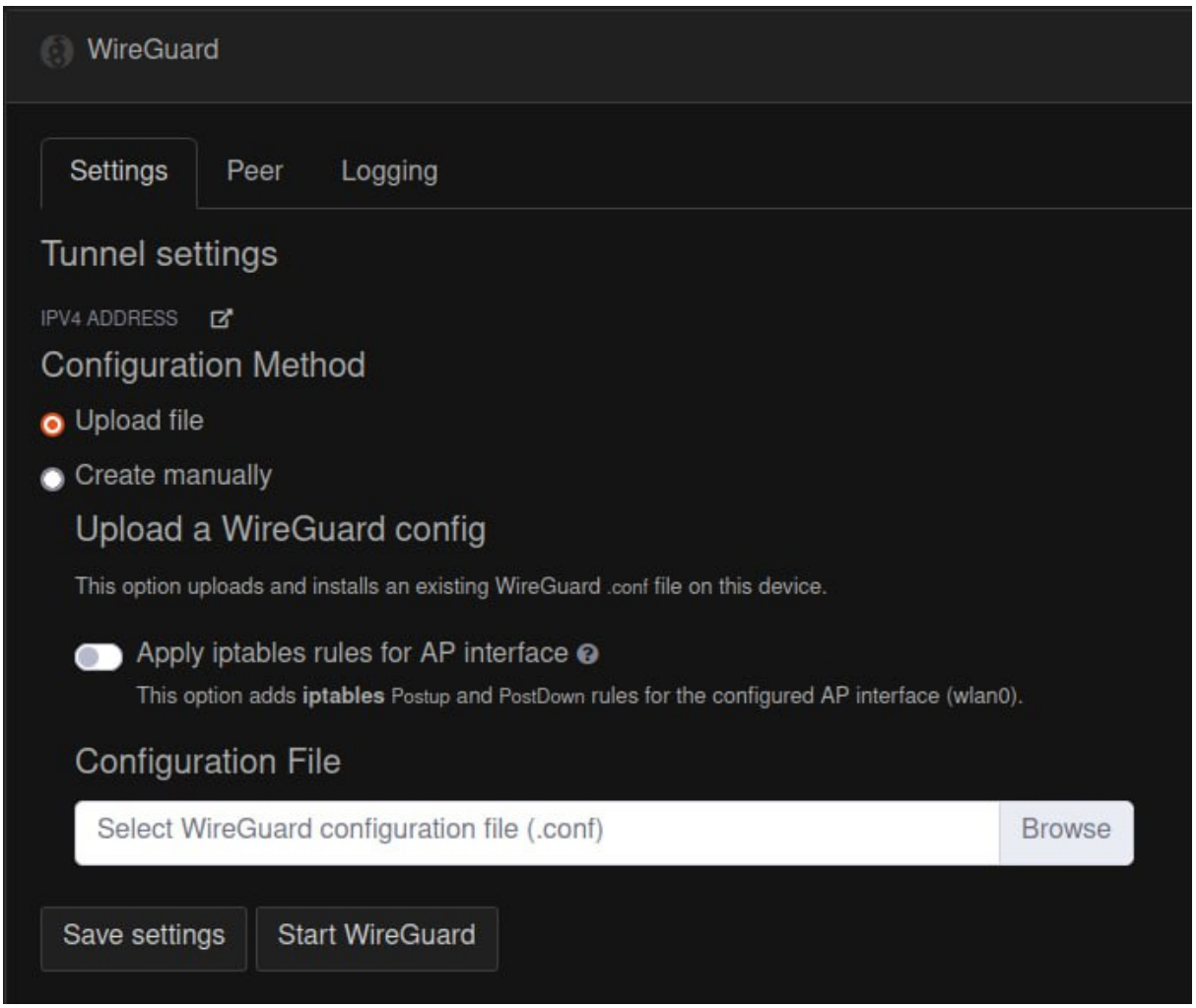


Рисунок 7.18 – Налаштування WireGuard

У розділі Peer ми маємо можливість додавати додаткові хости та їх публічні ключі шифрування. Ця функція дозволяє нам встановлювати зв'язок з іншими вузлами (хостами) у мережі WireGuard. Ми можемо вказати необхідні параметри, такі як IP-адреси та публічні ключі, для забезпечення безпеки та конфіденційності комунікації між різними вузлами. Це важливий аспект для налаштування безпечного з'єднання та забезпечення захисту наших даних під час їх передачі через мережу.

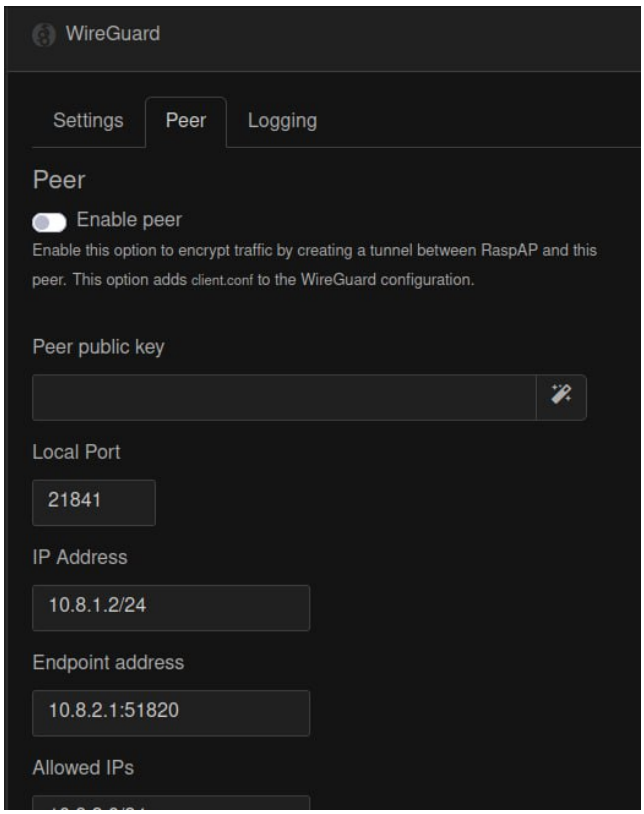


Рисунок 7.19 – Налаштування Peer для WireGuard

У розділі Logging ми зможемо переглянути журнали (логи) WireGuard. Ця функція дозволяє нам відстежувати події та діагностичну інформацію, пов'язану з роботою WireGuard. Журнали можуть містити записи про встановлення тунелей, зміни конфігурації, сповіщення про помилки або інші події, які стосуються роботи з VPN-протоколом WireGuard. Це дозволяє нам здійснювати моніторинг та аналіз роботи VPN-з'єднання для виявлення проблем або вирішення неполадок у нашій мережі.

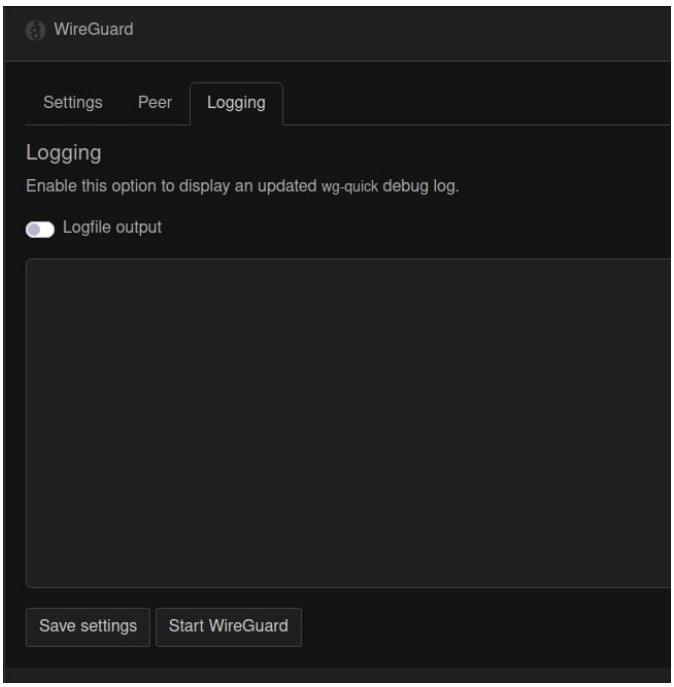


Рисунок 7.20 – Журнал подій WireGuard

7.7 Налаштування авторизайції до веб-інтерфейсу

У розділі Authentication ми маємо можливість змінити дані для авторизації у веб-інтерфейсі точки доступу Hotspot. Тут ми можемо змінити ім'я користувача (логін) або пароль, які використовуються для входу в систему. Ця функція дозволяє нам забезпечити безпеку нашого пристрою, змінюючи стандартні облікові дані та налаштовуючи власні дані для доступу до нашої точки доступу. Таким чином, ми можемо забезпечити контроль доступу до наших налаштувань і зберегти їх від несанкціонованого доступу.

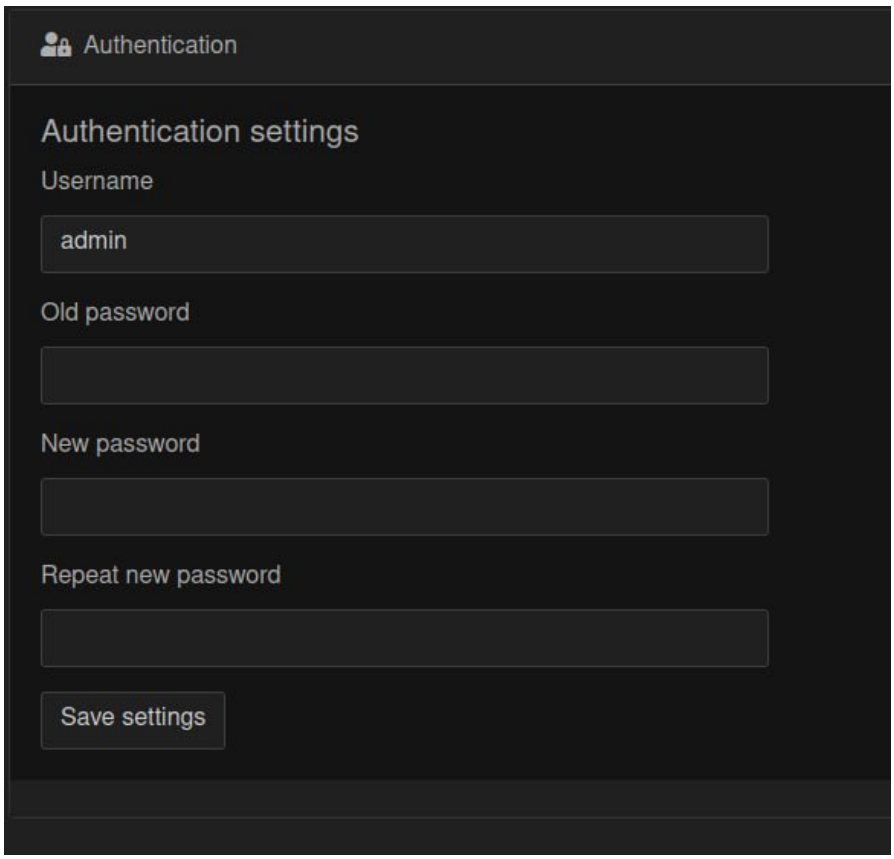


Рисунок 7.21 – Налаштування авторизації у WEB інтерфейс

7.8 Інформація про систему

У розділі System ми можемо переглянути інформацію про нашу систему, включаючи такі дані як HOST NAME, PI REVISION, OS, KERNEL, UPTIME. Також ми можемо переглянути статистику використання оперативної пам'яті (MEMORY USED), навантаження процесора (CPU LOAD) та температуру процесора (CPU TEMP). Крім того, у цьому розділі ми маємо можливість перезапустити точку доступу або вимкнути її, що дозволяє нам здійснювати контроль над роботою пристрою і виконувати необхідні дії для збереження його працездатності.

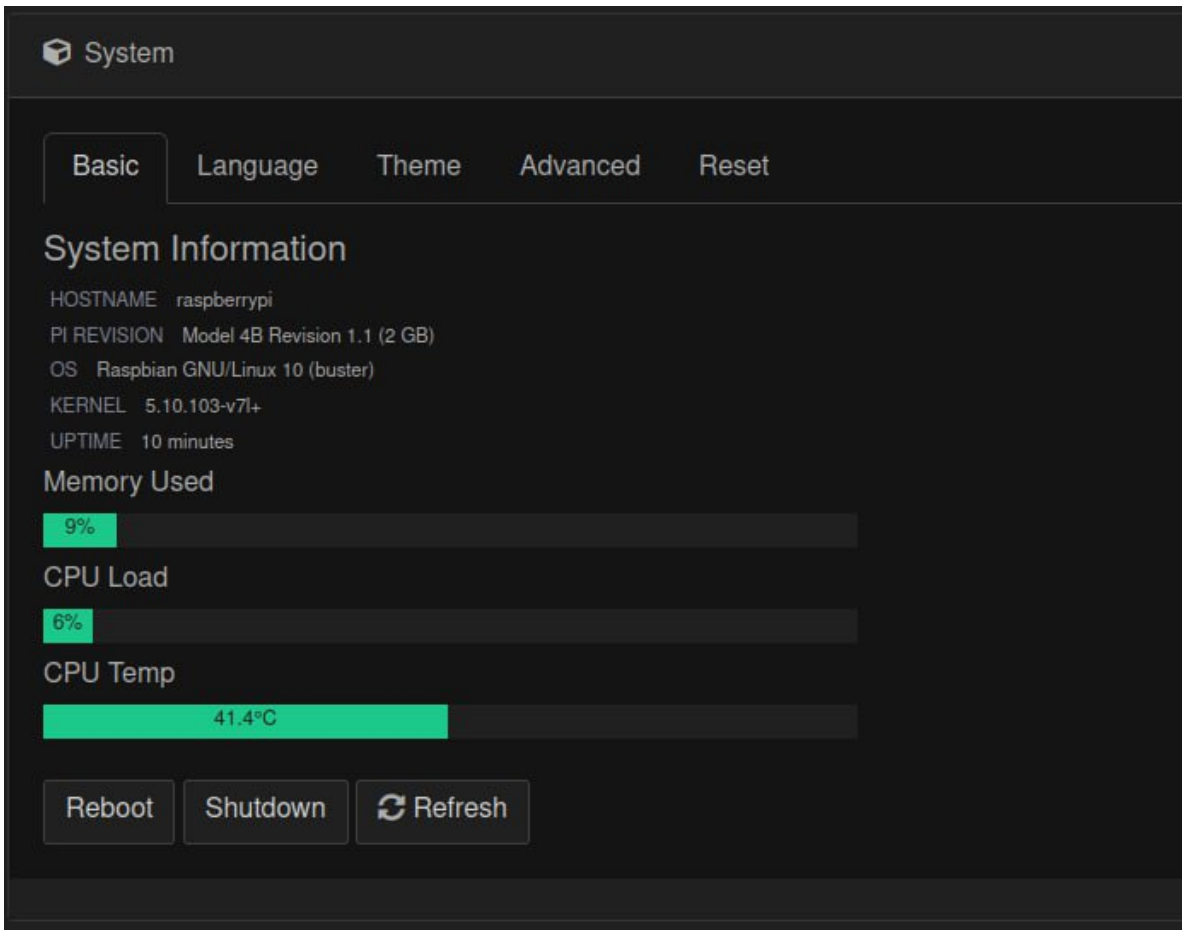


Рисунок 7.22 – Інформація про систему

У розділі Language ми маємо можливість змінити мову веб-інтерфейсу. Це дозволяє нам налаштувати інтерфейс у відповідності до наших вподобань та потреб, обираючи з доступних мовових опцій ту, яка найбільше відповідає нашим потребам. Зміна мови веб-інтерфейсу допомагає забезпечити зручну та зрозумілу навігацію та взаємодію з пристроєм.

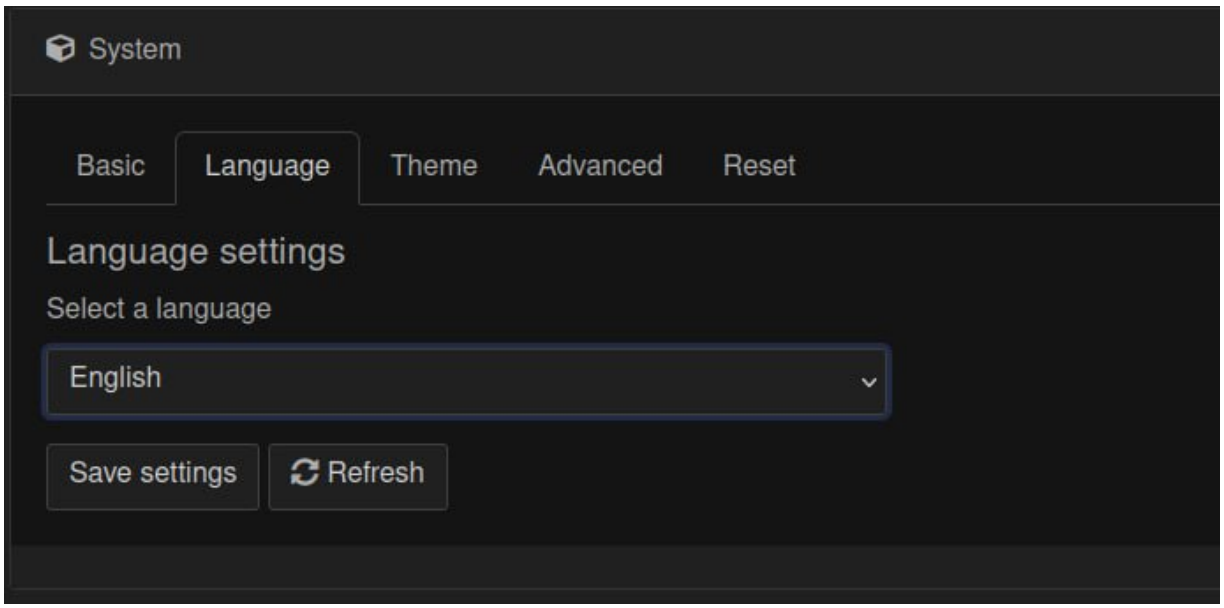


Рисунок 7.23 - Зміна мови інтерфейсу

В розділі Theme ми маємо можливість налаштувати зовнішній вигляд веб-інтерфейсу. Тут ми можемо змінити тему, вибрати бажаний колір шапки та логотип. Це дозволяє нам налаштувати інтерфейс так, щоб він відповідав нашим вподобанням і стилістиці. Зміна теми та кольорової схеми допомагає створити приємне і персоналізоване візуальне сприйняття під час використання веб-інтерфейсу.

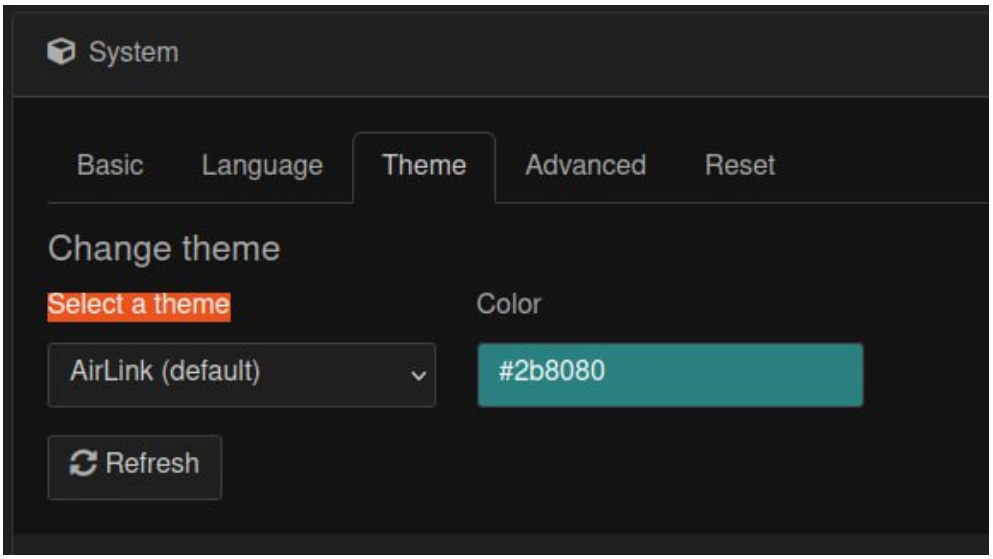


Рисунок 7.24 – Кастомізація WEB інтерфейсу

В розділі Advanced ми маємо можливість змінити порт веб-інтерфейсу та вказати IP-адресу, з якої буде доступний доступ до нього. Зміна порту дозволяє нам використовувати інший номер порту, щоб забезпечити більшу безпеку та унікальність доступу до інтерфейсу. Крім того, ми можемо вказати конкретну IP-адресу, з якої буде можливість підключення до веб-інтерфейсу, забезпечуючи контроль над доступом та забезпечуючи безпеку нашої системи.

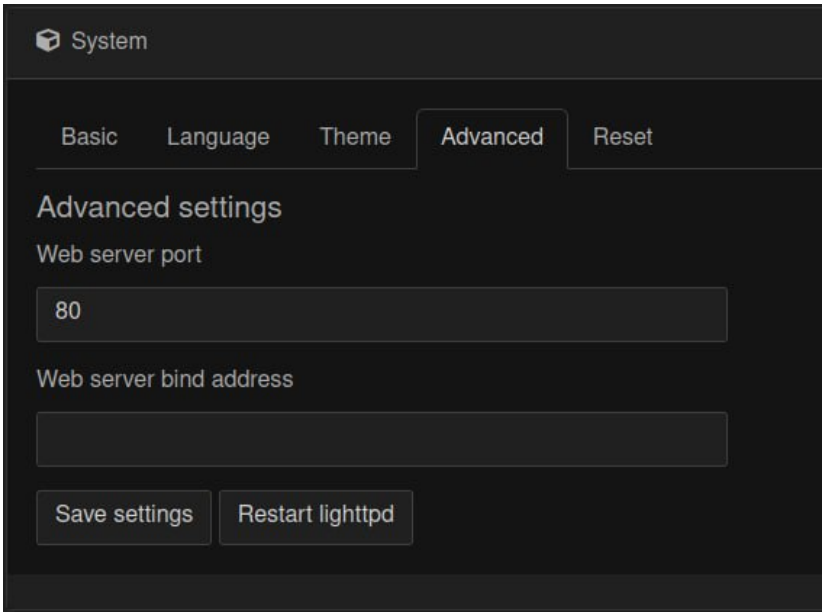


Рисунок 7.25 – Налаштування порту для підключення в WEB інтерфейс

У розділі Reset ми маємо можливість скинути всі налаштування на стандартні значення. Це включає скидання всіх змінених параметрів до їхніх початкових значень, що дозволяє повернути систему до початкового стану. Це корисна функція, яка може бути використана, наприклад, якщо ми хочемо почати спочатку з налаштуваннями або виправити помилки, які могли статися в процесі налаштування. Переконайтеся, що ви розумієте наслідки цієї дії, оскільки всі ваші поточні налаштування будуть втрачені після скидання до стандартних значень.

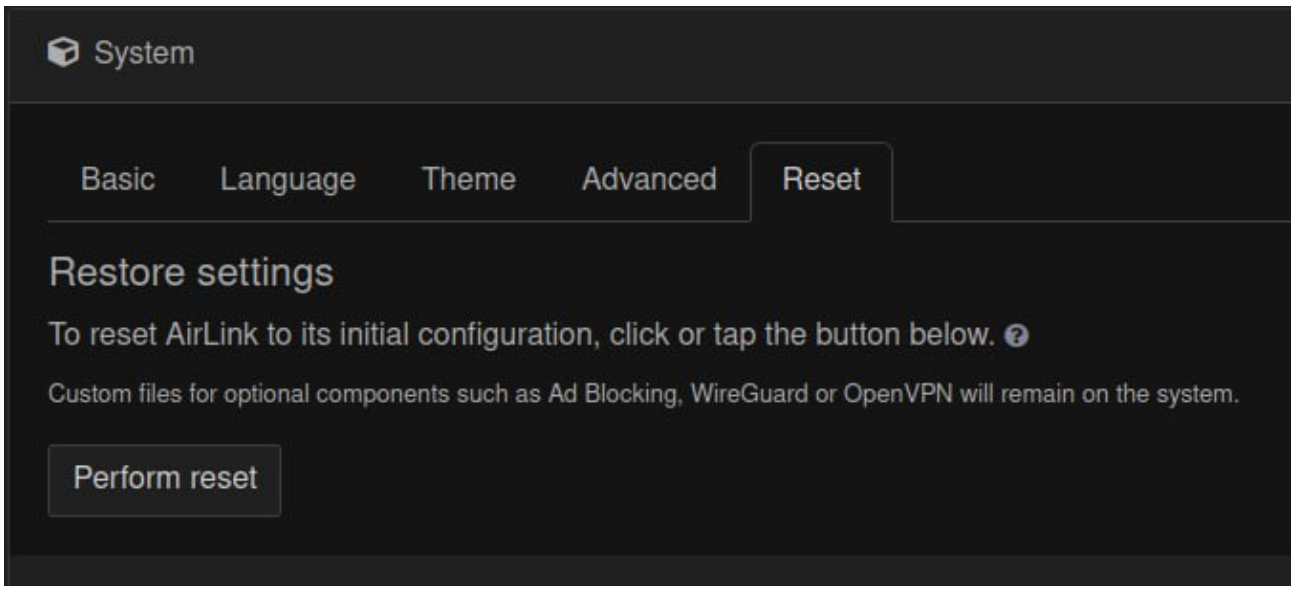


Рисунок 7.26 – Встановлення налаштування за замовчуванням

7.9 Облік трафіку

У розділі Data usage ми можемо переглянути статистику використання інтернет-трафіку за різні періоди часу, такі як день, неділя та місяць. Це дозволяє нам контролювати та відстежувати обсяг трафіку, який ми використовуємо нашою точкою доступу. Ця інформація може бути корисною для моніторингу нашої мережі, планування ресурсів та виявлення аномального використання трафіку. Ми можемо легко переглянути загальний обсяг трафіку, використаний протягом зазначеного періоду, що допомагає нам керувати та контролювати нашу інтернет-активність.

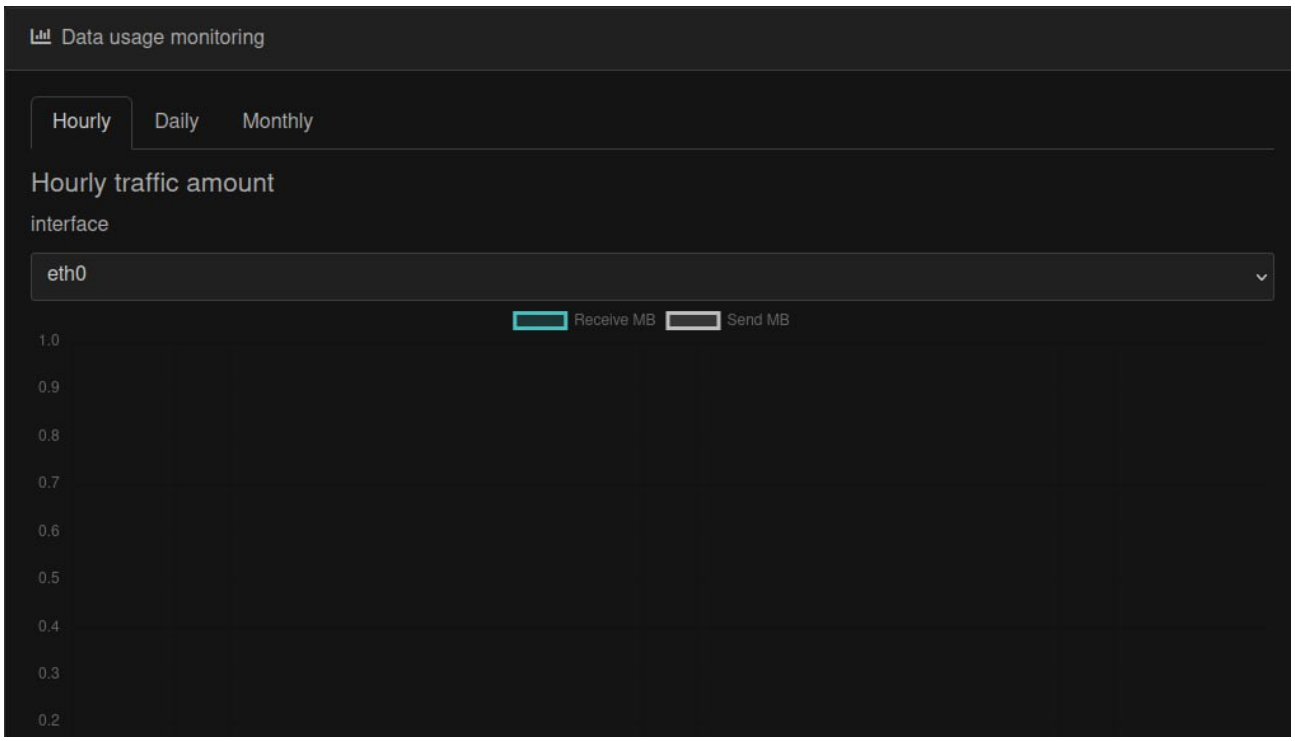


Рисунок 7.27 – Журнал використаного трафіка.

Інтерфейс hotspot зручний та легкий у використанні, що дозволяє користувачам з легкістю керувати своєю точкою доступу. Його простий та інтуїтивно зрозумілий дизайн дозволяє легко навігувати по різних розділах та налаштуваннях. Навіть для користувачів без технічних навичок, інтерфейс забезпечує зручну інтеракцію зі всіма функціями hotspot.

Користувачі можуть швидко знайти необхідні розділи та опції, такі як управління точкою доступу, налаштування безпеки, управління підключеними пристроями та перегляд статистики використання трафіку. Всі ці функції доступні в інтуїтивно зрозумілому форматі, де користувачі можуть легко виконувати потрібні дії та налаштування без зайвих зусиль.

Безперечно, інтерфейс є дружнім до користувача та спрощує процес керування hotspot. Він забезпечує зручність і простоту використання, що робить його ідеальним вибором для будь-якого користувача, незалежно від їх технічних навичок.

ВИСНОВКИ

У рамках дослідження була проведена розробка портативного хот-споту з функціями мережного екрану на базі Raspberry Pi. Цей проект дозволяє поєднати мобільність та потужність Raspberry Pi для створення зручного та функціонального пристрою.

Актуальність теми полягає в зростаючому попиті на портативні хот-споти та необхідності забезпечення безпеки та контролю доступу до Інтернету. Розроблений портативний хот-спот на базі Raspberry Pi має потенційні переваги, такі як гнучкість, доступність та можливість розширення функціональності.

Завдання дослідження включали вибір апаратного забезпечення, розробку програмного забезпечення, налаштування мережевих функцій та інтерфейсу користувача, а також тестування та оцінку пристрою. У результаті було розроблено пристрій, який може надати надійний та безпечний доступ до Інтернету в будь-якому місці, забезпечуючи контроль доступу та захист від шкідливого контенту.

Отже, розробка портативного хот-споту з функціями мережного екрану на базі Raspberry Pi є значимим кроком у напрямку поліпшення доступу до Інтернету та забезпечення безпеки користувачів. Цей пристрій має потенціал для застосування в різних сферах, включаючи мобільний Інтернет, подорожі, роботу віддалено та багато іншого. Результати дослідження відкривають можливості для подальшого розвитку та вдосконалення портативних хот-спотів з функціями мережного екрану на базі Raspberry Pi.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 PHP documentation – Режим доступу: https://www.php.net/docs.php

2 Raspberry Pi documentation – Режим доступу: https://www.raspberrypi.com/documentation/

3 Python documentation – Режим доступу: https://docs.python.org/3/

4 Framework scapy documentation – Режим доступу: https://scapy.readthedocs.io/en/latest/

5 Hostapd documentation – Режим доступу: https://wireless.wiki.kernel.org/en/users/documentation/hostapd

6 Dnsmask documentation – Режим доступу: https://thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html

7 HTML documentation – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/HTML

8 CSS documentation – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/CSS

9 JavaScript documentation – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/JavaScript

10 Linux documentation – Режим доступу: https://tldp.org/LDP/sag/html/index.html

11 Snort documentation – Режим доступу: https://www.snort.org/documents

12 Lighttpd documentation – Режим доступу: https://redmine.lighttpd.net/projects/1/wiki/docs

13 Основи комп`ютерних мереж – Режим доступу: https://stud.com.ua/120704/informatika/osnovi_pobudovi_kompyuternih_merezh