НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Радіотехнічний факультет

Кафедра радіоінженерії

«На правах рукопису»
УДК _____

До захисту допущено:
Завідувач кафедри
_____ Сергій МАРТИНЮК
«___»_____2022р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Інформаційна та комунікаційна радіоінженерія»

зі спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Програмно-апаратний комплекс протоколювання та сповіщення про атаку на внутрішню мережу»

Виконав (-ла):
студент (-ка) ІІ курсу, групи РІ-11мп
Козик Антон Владиславович _____

Керівник:
К.т.н., доц. кафедри РІ
Михайленко Максим Вікторович _____

Рецензент:
Ст.викл каф. РТС
Турєєва Ольга Василівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2022 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Радіотехнічний факультет

Кафедра радіоінженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма - «Інформаційна та комунікаційна радіоінженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій МАРТИНЮК

«__5_»__вересня____2022р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Козику Антону Владиславовичу

1. Тема дисертації: «Програмно-апаратний комплекс протоколювання та сповіщення про атаку на внутрішню мережу_____
_____
_____», науковий керівник дисертації Михайленко Максим Вікторович, к.т.н., доц.,

затверджені наказом по університету від «__09_»___листопада__ 2022 р. №__4137-с___

2. Термін подання студентом дисертації: 12.12.2022_____
3. Об'єкт дослідження: захист внутрішньої мережі від атак_____
_____
_____

4. Вихідні дані: інформація про потенційно небезпечні мережеві порти і способи захисту внутрішньої мережі_____
_____
_____

5. Перелік завдань, які потрібно розробити: провести дослідження способів проникнення у внутрішню мережу, провести дослідження небезпечних портів, створити програмно-апаратний комплекс сповіщення атаки на внутрішню мережу_____

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: 38 рисунків, 2 таблиці

7. Орієнтовний перелік публікацій: 19 найменувань у переліку посилань

8. Дата видачі завдання 5 вересня 2022р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Дослідження актуальності теми, способів проникнення у внутрішню мережу	05.09 – 11.09	Виконано
	Пошук інформації та літератури стосовно видів атак та захисту від них.	12.09 – 21.09	Виконано
	Визначення портів, які найчастіше стають мішенню для кіберзлочинців. Пошук інформації про призначення портів	22.09 – 04.09	Виконано
	Складання вимог до програмно-апаратного комплексу, опис функціоналу	05.09 – 11.10	Виконано
	Вибір підходящох техніки для програмно-апаратного комплесу, порівняння варіантів	12.10 – 17.10	Виконано
	Робота над кодом M5Stack, обробка TCP та UDP пакетів	17.10 – 28.10	Виконано
	Розробка частини серверного коду, обробка POST запитів за допомогою PHP скріпта	29.10 – 16.11	Виконано
	Створення PHP скріпта, для відображення вмісту бази даних в браузері	16.11 – 24.11	Виконано
	Корегування текстової частини диплому, виправлення помилок та деякі зміни в структурі плану	25.11 – 11.12	Виконано

Студент Антон КОЗИК

Науковий керівник Максим МИХАЙЛЕНКО

АНОТАЦІЯ

Пояснювальна записка дипломної роботи має обсяг 87 сторінок, містить 38 рисунків та ілюстрацій, 7 таблиць, 19 найменувань у переліку посилань.

Актуальність теми обумовлена збільшенням числа кібератак на організації та окремих осіб, через цифровізацію та все більше число віддалених видів робіт.

Метою даної роботи є створити програмно-апаратний комплекс сповіщення про атаку на внутрішню мережу.

Увесь текст поділено на 5 послідовних глав, у яких зроблений огляд відомих типів кібератак, проаналізовані їх особливості, показаний спільний етап більшості кібератак - сканування портів, розглянуті особливості найбільш вразливих портів, через які зловмисники найчастіше проводять атаки. Досліджено такі об'єкти як мережевий порт, сканування портів, проброс портів, трансляція мережевих портів (NAT). Наведені графічні матеріали, які демонструють як змінюється вміст пакету під час його передачі в мережі.

На основі проведеного огляду типів кібератак та аналізу їх особливостей, запропонований алгоритм виявлення кібератак, та розроблені вимоги до програмно-апаратного комплексу виявлення та протоколювання кібератак на внутрішню мережу.

На основі цих вимог складено ТЗ. Проведений та обгрунтований вибір апаратної частини комплексу, у якості якої обраний мікроконтролер M5Stack. Також описано процес створення програм для апаратних засобів комплексу. Створені програми наведені у додатках до роботи.

Відмінною рисою роботи є створення діючого прототипу розробки, іспит створенного прототипу та надання результатів іспиту. Крім того, представлені фото прототипу та скірншоти із результатами роботи приладу. Робота буде корисна тим, хто хоче отримати інформацію про досягнутий на сьогодні рівень розвитку кібербезпеки та способів і засобів захисту від

кіберзлочинних атак.

Ключові слова: мікроконтролер, M5Stack, кібербезпека, кібератаки, сканування портів, програмування.

Ключові слова: мікроконтролер, M5Stack, кібербезпека, кібератаки, програмування.

ANNOTATION

The relevance of the topic is due to the increase in the number of cyber attacks on organizations and individuals, due to digitalization and an increasing number of remote work.

The purpose of this work is to create a software and hardware complex for notification of an attack on an internal network. The entire text is divided into 5 consecutive chapters, which provide an overview of the known types of cyber attacks, analyze their features, show the common stage of most cyber attacks - port scanning, consider the features of the most vulnerable ports through which attackers most often conduct attacks. Such objects as network port, port scanning, port forwarding, network port translation (NAT) are investigated. Graphic materials are provided that demonstrate how the contents of a packet change during its transmission in the network.

Based on the review of the types of cyberattacks and analysis of their features, the algorithm for detecting cyberattacks is proposed, and the requirements for the software and hardware complex for detecting and logging cyberattacks on the internal network are developed.

On the basis of these requirements, a technical specification was drawn up. The choice of the hardware of the complex was made and justified, as which the M5Stack microcontroller was chosen. The process of creating programs for the hardware of the complex is also described. The created programs are given in the appendices to the work. A distinctive feature of the work is the creation of a working prototype of the development, the examination of the created prototype and the presentation of the examination results. In addition, photos of the prototype and screenshots with the results of the device are presented. The work will be useful for those who want to get information about the current level of development of cybersecurity and ways and means of protection against cybercrime attacks.

Keywords: microcontroller, M5Stack, cybersecurity, cyberattacks, port scanning, programming.

ЗМІСТ

Зміст 6

Перелік скорочень 9

Вступ 10

1 Кібербезпека 11

1.1 Актуальність теми 11

1.2 Задачі злому та шляхи реалізації задач 12

1.3 Задача захисту від проникнення у внутрішню мережу 14

2 Дослідження 16

2.1 Способи проникнення у внутрішню мережу 16

2.1.1 Сканування портів 17

2.1.2 Атака перебором автентифікаційних даних (брутфорс) 20

2.1.3 Анонімна автентифікація 24

2.1.4 Атака NAT slipstreaming 25

2.1.5 Комбінація DHCP starvation і Rogue DHCP Server 28

2.2 Порти та їх призначення 29

2.2.1 Теоритичні відомості про мережеві порти 29

2.2.2 Ризики безпеки, пов’язані з портами 30

2.2.3 Вразливі порти, які потребують вашої уваги 31

2.2.4 Дослідження Alert Logic уразливості портів 33

2.3 Сценарії протидії злому 35

3 Постановка завдання на розробку 36

3.1 Вимоги до програмно-апаратного комплексу. Функціонування ... 36

3.2 Вимоги до апаратної частини 37

4 Розробка 39
4.1 Апаратна частина 39
4.1.1 Arduino Uno 39
4.1.2 Raspberry Pi Zero 41
4.1.3 Nodemcu 41
4.1.4 Beaglebone 42
4.1.5 ESP8266 43
4.1.1 Опис структури M5Stack 45
4.1.1 Підсумок 47
4.2 Програмна частина 48
4.2.1 Налагодження роутера / шлюза у зовнішню мережу 48
4.2.2 Розробка частини програми сповіщення про атаки 52
4.2.3 Розробка частини програми протоколювання 56
5 Іспит програмно-апаратного комплексу 62
5.1 Програма іспиту 62
5.2 Обладнання іспиту 63
5.3 Результати іспиту 64
6 Розроблення стартап-проєкту 70
6.1 Маркетинговий аналіз стартап-проєкту 70
6.2 Розроблення маркетингової програми стартап-проекту 72
6.3 Висновки до стартап-проєкту 73
Висновки 74
Перелік літератури 76
Додаток 1 79
Додаток 2 80

Додаток 3 82

Додаток 4 84

ПЕРЕЛІК СКОРОЧЕНЬ

ARP — Address Resolution Protocol.

ICMP — Internet Control Message Protocol.

FTP — File Transfer Protocol.

SSH — Secure Shell.

UDP — User Datagram Protocol.

TCP — Transmission Control Protocol.

ALG — Application-level gateway.

NAT — Network Address Translation.

HTTP — HyperText Transfer Protocol.

MitM — Man In The Middle.

DHCP — HyperText Transfer Protocol.

SQL — Structured query language.

DoS — Denial-of-service attack.

DdoS — Distributed Denial-of-Service.

БД — База даних.

ВСТУП

Кібербезпека є важливою складовою нашого життя, особливо в умовах, коли світ стає все більш цифровим. Ті, хто не інвестував у належні рішення з кібербезпеки у 2022 році, будуть у суттєво невигідному становищі, оскільки очікується, що частота та складність кібератак лише зростатимуть.

Хоча важливість кібербезпеки вже була очевидна протягом тривалого часу, останні роки знову привернув увагу до цього питання. Пандемія прискорила тенденцію цифрової трансформації, оскільки компанії звернулися до віддаленої роботи та онлайн-рішень, щоб утриматися на плаву. У той же час цифрова трансформація також створила нові можливості і для кіберзлочинців у напрямку атак окремих осіб і компаній.

Пандемія змусила бізнес використовувати дистанційну та гібридну роботу, і очікується, що ця тенденція збережеться і надалі. Нещодавнє опитування Gallup серед понад 140 000 співробітників США показало, що 24% працюють виключно віддалено, тоді як 53% мають певну форму гібридної роботи.

Зрозуміло, що багато співробітників звикли працювати вдома, оскільки це дозволяє краще збалансувати роботу та особисте життя, але це також створює ризики для безпеки. По-перше, віддалені працівники частіше використовують особисті пристрої для роботи, що може призвести до витоку даних, якщо ці пристрої не мають відповідного програмного забезпечення безпеки. Існує також проблема використання незахищених Wi-Fi мереж, через що підприємства можуть бути вразливими до кібератак.

У дипломній роботі з метою зменшення ризику кібератак на основі M5Stack створено програмно-апаратний комплекс попередження про атаки на внутрішню мережу, розглянуто види можливих атак, проаналізовано слабкі місця у системі і способи мінімізації шкоди. Крім цього, було досліджено способи проникнення у внутрішню мережу, види портів та їх призначення, а також сценарії протидії злому.

1 КІБЕРБЕЗПЕКА

1.1 Актуальність теми

Кібербезпека — це практика захисту мереж, пристроїв і даних від пошкодження, втрати або несанкціонованого доступу. Так само, як фізична безпека захищає людей та будівлі від різного роду фізичних загроз, кібербезпека захищає технології та їх користувачів від небезпек цифрових.

Кібербезпека охоплює багато різних дисциплін, дій, загроз та ідей. Однак всі ці частини зводяться до однієї основної ідеї: захисту цифрового життя та активів людей. Такі речі, як цифрова валюта, персональні дані та доступ до деяких комп'ютерів, є основними цілями для злочинців, тому їх захист є надзвичайно важливим.

Наше повсякденне життя більше, ніж будь-коли раніше, залежить від технологій. Переваги цієї тенденції варіюються від майже миттєвого доступу до інформації в Інтернеті до сучасних зручностей, які забезпечують технологія автоматизації розумного дому та такі концепції, як Інтернет речей.

Проте кожен пристрій несе в собі приховану загрозу. Постійне зростання кіберзлочинності підкреслює недоліки пристроїв і послуг, від яких ми звикли залежати.

За даними статті Forbes [1] існує ризик збільшення кількіості кібератак та їх впливу на наше життя, а паралельно із цим, вірогідно, зменшиться кількість талановитих людей у сфері кібербезпеки через розвиток штучного інтелекту, який вже невдовзі зможе виявляти загрози не менш ефективно, ніж люди. Однією з причин збільшення кібератак, автор статті, називає збільшення кількості пристроїв IoT. Часто такі пристрої є мінімально захищеними, через що і стають легкою мішенню для кіберзлочинців. Найбільш вагомою проблемою кібербезпеки, на мою думку, є те що наслідки дій хакерів можуть бути більш, ніж реальними та відчутними фізично, мова йде про зломи

компонентів інфраструктури, таких як лікарні, транспортні системи або навіть електростанції чи греблі.

Відповідно до статті журналу Cybercrime Magazine [2], до 2025 року кіберзлочинність коштуватиме світу 10,5 трильйонів доларів США щорічно. Крім того, прогнозується, що глобальні витрати на кіберзлочинність зростатимуть майже на 15 відсотків щороку протягом наступних чотирьох років. Також в цій статті вказується, що кіберзлочинність настільки сильно вдарила по США, що в 2018 році спеціальний агнет ФБР, який розслідує кібервторгення, дав коментар The Wall Street Journal, що коже американський громадянин повинен очікувати, що всі його дані (особиста інформація) були викрадені та знаходяться в даркнеті.

1.2 Задачі злому та шляхи реалізації задач

Які ж цілі переслідують кіберзлочинці? Більше 60% [3] це фінансові мотиви.

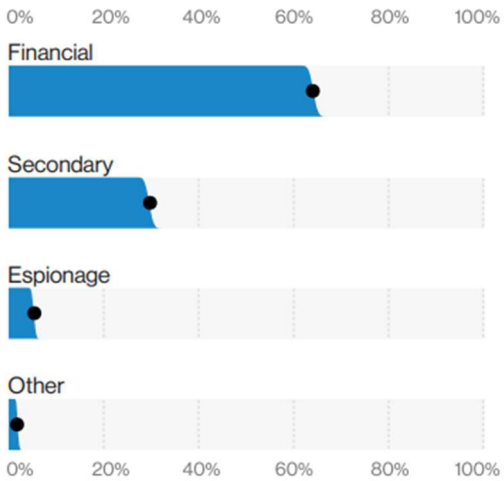

Рисунок 1.1 — Мотиви здійснення кібератак на основі 3,828 [3]

Наслідки говорять самі за себе.

Наприклад:

- Середній час простою бізнесу через атаки програм-вимагачів становить 22 дні [4].

- Організації, які постраждали від шахрайських атак, втрачають від 25 000 до 75 000 доларів США [5].

В основному хакери націлені на 4 основних галузі:

1. **Медичні заклади.** Кібератаки в сфері охорони здоров'я можуть бути різними, наприклад відмова в обслуговуванні, шкідливий код та програми-вимагачі. Ці атаки негативно впливають не лише на медичних працівників, а й на їхніх пацієнтів [6]:
    - 2020 рік: постраждали 34 мільйони пацієнтів.
    - 2021 рік: постраждало 45 мільйонів пацієнтів.

2. **Уряд.** Державні організації володіють великою кількістю конфіденційної інформації, такої як відбитки пальців, SSN тощо.
    - Середній розмір викупу за вкрадені дані в державних установах становить 570 857 доларів США [7].
    - Понад $1,75 млн було виплачено хакерам у 2020 році [8].
    - 53,2% кібератак в органах державної влади спрямовані на міста та місцеві школи [9].

3. **Неурядові організації.** Половина НУО зазнали атак з використанням програм-вимагачів [10]. НУО є особливо вразливими та ідеальними цілями через їх фінансові дані. Такі ресурси, як портали для пожертвувань або збору коштів, слугують основними джерелами конфіденційної та грошової інформації. В прикалд можна привести нещодавні атаки на організації такого типу:
    - У січні 2022 року було зламано сервер "Лікарів без кордонів" в Іспанії [11]. Брокери Acess розмістили оголошення на форумі в даркнеті, пропонуючи вхід на один із серверів

організації. На щастя, вдалося уникнути серйозного витоку даних, але потенційна вразливість даних неприбуткової організації все одно викликає занепокоєння.

- У лютому 2022 року хакери атакували Міжнародний комітет Червоного Хреста (МКЧХ) [12], використовуючи код, призначений для ураження серверів. Якби не було вжито проактивних заходів, це могло призвести до втрати даних про майже 500 000 осіб.

4. **Фінанси та страхування.** Фінансова індустрія надає безліч можливостей для отримання максимального прибутку та впливу - двох критеріїв, які є важливими для досягнення цілей злочинців. Саме тому індустрія фінансових послуг частіше стає жертвою кібератаки, ніж інші організації. Крім того, дослідження компанії IBM X-Force [13] свідчить, що
    - 70% цифрових атак на фінансові та страхові фірми спрямовані на банки.
    - 16% атакують страхові організації.
    - 14% атакують інші фінансові підприємства.

1.3 Задача захисту від проникнення у внутрішню мережу

Зважаючі на те, що абсолютно для всіх із вищеперераховних випадків критичним є факт потрапляння зловмисника у внутрішню мережу, який підвищує його шанси в рази, її захист я й обрав для себе за ціль.

У першу чергу задачею є не дати зловмиснику зламати нашу мережу, сповістити про спробу отримати доступ, а також зберегти дані про те, хто намагався здійснити злом і вжити заходів для того, аби така атака не повторилася знову. Сповіщення буде мати вигляд повідомлення на дисплеї пристрою із даними про порти, на які було здійснено атаку і про IP-адресу, з якої прийшов запит. Ці дані будуть збережені до спеціального протоколу, який

в майбутньому може бути проаналізовано. Мірами, які можуть бути прийняті є блокування користувача у мережі по IP-адресі або ж можна просто дати можливість оператору, який буде слідкувати за безпекою мережі, самому вирішити які дії слід прийняти.

2 ДОСЛІДЖЕННЯ

2.1 Способи проникнення у внутрішню мережу

На сьогоднішній день існує безліч видів атак, здійснення яких можливе через відкриті порти. Якщо сервіс потребує автентифікації, зловмисник може здійснити атаку за словником або брутфорс для доступу до сервісу. Також він може здійснити атаку відмови у доступі, щоб служба перестала працювати належним чином і припинила підтримувати обслуговування.

Таблиця 2.1 — способи проникнення у внутрішню мережу

№	Спосіб проникнення
1	Пряме підключення до лінії зв'язку законного користувача
2	Проникнення до чужої інформаційної мережі шляхом автоматичного перебору
3	Використання чужих паролів (видавання себе за законного користувача)
4	Підміна даних — заміна або введення нових даних, що здійснюються, як правило, коли інформація вводиться або виводиться
5	«Троянський кінь» — приховане введення в чужу систему команд, які допомагатимуть їй здійснити нові, незаплановані функції при одночасному зберіганні її працездатності
6	«Логічна бомба» — модифікація програм шляхом прихованого впровадження до програми набору команд, що повинні спрацювати за певних умов через деякий час
7	Здійснення доступу до баз даних і файлів законного користувача за рахунок знаходження слабких місць у системах захисту
8	Використання помилок у логіці побудови програми і виявлення «прогалин»

9	Поширення комп’ютерними мережами або продаж програм, що призводять до знищення або блокування інформації, порушення працездатності ЕОМ, системи ЕОМ або їх мережі, включаючи переважну більшість комп’ютерних вірусів

Кількість реалізацій таких атак складно перерахувати, включаючи те, що вона росте із кожним днем [14]: перевищення розміру фрагменту, «смертельний» запит, конфлікт протоколу ARP, синхронна атака, фрагменти, що перекриваються, атака прориву, атака маленькими фрагментами, фрагментований протокол управління групами Інтернет (IGMP), атака на папку локальної мережі (LAND Attack) і ін. Далі розглянуто деякі атаки більш детально.

2.1.1 Сканування портів

Зазвичай перед самим актом нападу хакери проводять дослідження мережі, на яку планується атака. Одним із методів таких досліджень є сканування портів.

Сканування портів — це поширена техніка, яку хакери використовують для виявлення відкритих дверей або слабких місць у мережі. Атака зі скануванням портів допомагає кіберзлочинцям знаходити відкриті порти та з’ясувати, чи вони отримують та надсилають дані. Сканування також може виявити, чи використовуються в організації активні пристрої безпеки, такі як брандмауери.

Коли хакери надсилають повідомлення до порту, відповідь, яку вони отримують, визначає, чи використовується порт і чи є потенційні недоліки, якими можна скористатися.

```
anthoz@DESKTOP-J52FO8A: ~
anthoz@DESKTOP-J52FO8A:~$ sudo nmap 192.168.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-15 19:36 EEST
Nmap scan report for Dlink-Router.Dlink (192.168.0.1)
Host is up (0.0073s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
8888/tcp open  sun-answerbook
9999/tcp open  abyss

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
anthoz@DESKTOP-J52FO8A:~$
```

Рисунок 2.1 — приклад використання мережевого сканеру Nmap

Деякі компанії самі також можуть використовувати техніку сканування портів, щоб надсилати пакети на певні порти та аналізувати відповіді на наявність потенційної вразливості. Вони можуть використовувати такі інструменти, як IP-сканування, мережевий сканер (Nmap) (рис. 2.1) і Netcat (рис. 2.2), щоб забезпечити безпеку своєї мережі та систем.

```
anthoz@DESKTOP-J52FO8A: ~
anthoz@DESKTOP-J52FO8A:~$ nc -z -v 192.168.0.1 1234 23 80 8080
nc: connect to 192.168.0.1 port 1234 (tcp) failed: Connection refused
Connection to 192.168.0.1 23 port [tcp/telnet] succeeded!
Connection to 192.168.0.1 80 port [tcp/http] succeeded!
nc: connect to 192.168.0.1 port 8080 (tcp) failed: Connection refused
anthoz@DESKTOP-J52FO8A:~$
```

Рисунок 2.2 — приклад використання мережевого сканеру Netcat

Сканування портів може надати таку інформацію, як:

- Перелік працюючих сервісів
- Перелік користувачів, які володіють послугами
- Наявність входів, які не потребують автентифікації
- Для яких мережевих служб потрібна автентифікація

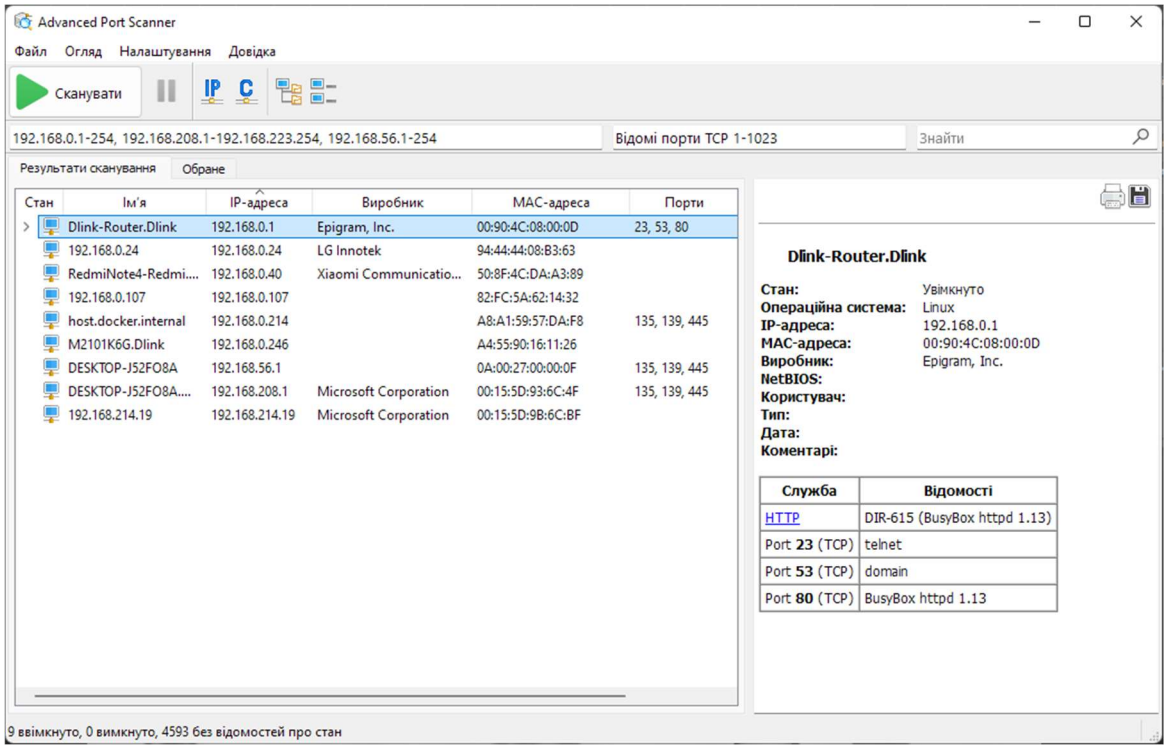

Рисунок 2.3 — приклад роботи сканера та інформація, яку він може надати

Існують різні способи сканування портів:

1. **Ping-сканування**: Ping-сканування вважається найпростішим методом сканування портів. Воно також відоме, як Internet Control Message Protocol (ICMP). Сканування Ping надсилає групу з кількох запитів ICMP на різні сервери в спробі отримати відповідь. Сканування ping може використовуватися адміністратором для усунення проблем, а ping може блокуватися та вимикатися за допомогою брандмауером.
2. **Vanilla scan**: ще одна базова техніка сканування портів, сканування vanilla намагається підключитися до всіх 65,536 портів одночасно. Воно надсилає позначку синхронізації (SYN) або запит на підключення. Коли програма отримує відповідь SYN-ACK або підтвердження підключення, вона відповідає прапором ACK. Це сканування є точним, але його легко виявити, оскільки повне з'єднання завжди реєструється брандмауерами.

3. **SYN-сканування**: також називається напіввідкритим скануванням, це сканування відправляє прапорець SYN до цілі і чекає відповіді SYN-ACK. У разі отримання відповіді сканер не відповідає, що означає, що TCP-з'єднання не було завершено. Таким чином, взаємодія не реєструється, але відправник дізнається, чи відкритий порт. Це швидкий прийом, який використовують хакери для пошуку слабких місць.
4. **XMAS та FIN-сканування**: сканування ялинкою (XMAS-сканування) та FIN-сканування є більш дискретними методами атаки. Сканування XMAS отримало свою назву від набору прапорів, які включаються в пакеті, і які при перегляді в аналізаторі протоколів, такому як Wireshark, виглядають як ялинка, що блимає. Цей тип сканування посилає набір прапорів, при відповіді на які, розкривається інформація про брандмауер і стан портів.
5. **Sweep scan:** цей метод попереднього сканування портів надсилає трафік до порту через комп'ютери у мережі для виявлення активних пристроїв. Він не передає жодної інформації про активність порту, але інформує відправника про те, чи використовуються якісь системи.

2.1.2 Атака перебором автентифікаційних даних (брутфорс)

Атака брутфорсом є однією із найпримітивніших, вона полягає в звичайній підстановці даних із готового словника із розповсюдженими паролями/логінами у відповідні рядки.

Даний тип атаки може бути застосований до TELNET, AFP, CVS, FTP, HTTP, HTTPS, SOCKS5, HTTP-PROXY, IMAP, MS-SQL, а також багатьох інших. Принцип роботи для кожного із випадків не практично не відрізняється, тому для прикладу я візмему Telnet та продемонструю реалізацію такої атаки.

Telnet (teletype network) — це мережевий протокол для реалізації текстового інтерфейсу по мережі, тобто для створення віддалених сеансів між пристроями. Часто його використовують для конфігурування роутерів.

Даний протокол за замовчуванням використовує 23 порт і є доступним.

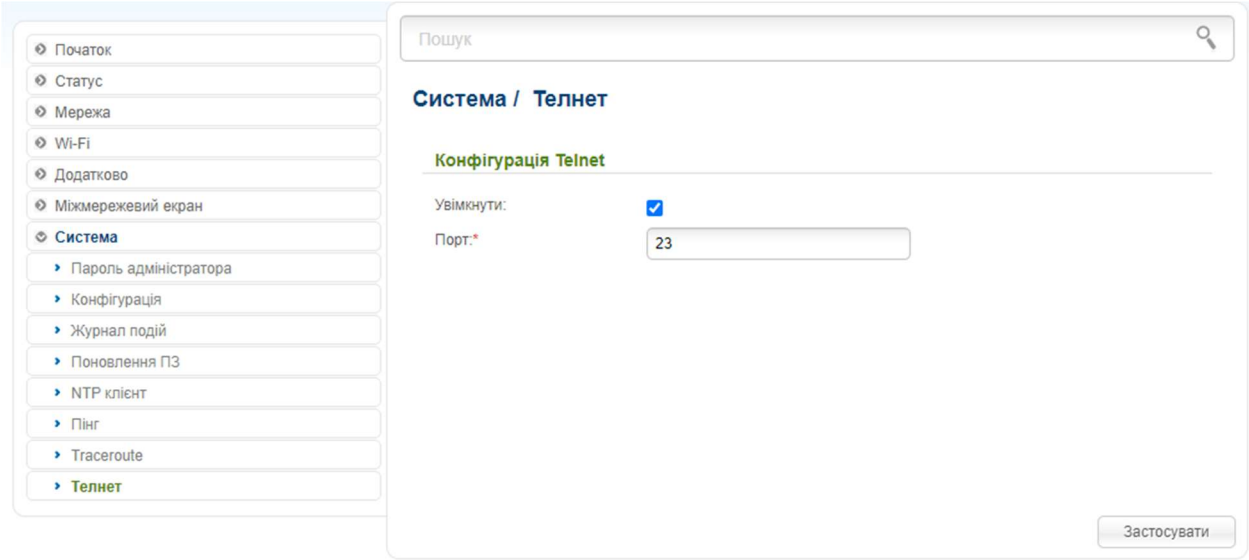

Рисунок 2.4 — Налаштування Telnet на D-Link DIR-615

Для того, щоб отримати повний доступ до конфігурацій маршрутизатора, достатньо виконати команду telnet <ip-address> <port> із вашого термінала (рис. 2.5).

Проблема полягає в тому, що часто користувачі не міняють стандартні логін та пароль (admin/admin), через це зловмиснику не складає ніякої проблеми проникнути у вашу локальну мережу.

Але навіть зміна паролю зовсім не гарантує повної безпеки, тому що Telnet не забезпечує захисту від простого перебору автентифікаційних даних. Хотілось би також відмітити, що для більшої надійності краще створити нового користувача із іншим логіном, а стандартного користувача admin видалити, це не дозволить хакеру швидко вгадати наші дані.

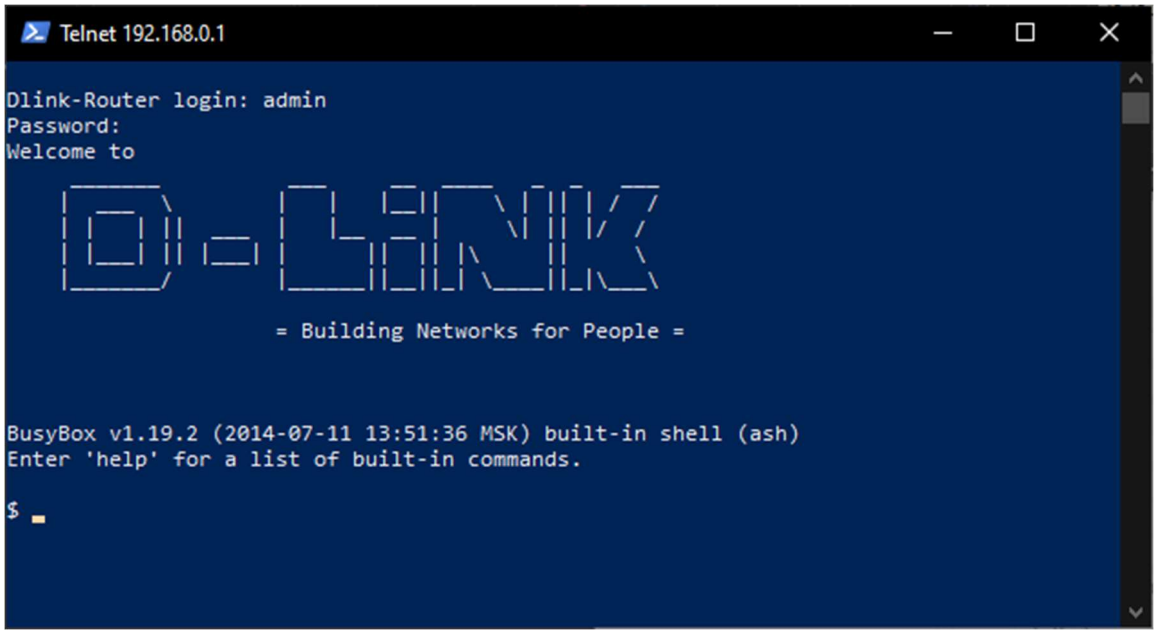

Рисунок 2.5 — під'єднання до маршрутизатору за допомогою Telnet

Для того, щоб продемонструвати як працює атака перебором було написано невелику програму на Python (додаток 1) яка буде перебирати паролі із файлу та намагатися увійти всередину роутера через Телнет.

Встановимо у якості пароля "mytestpassword" (рис. 2.6)

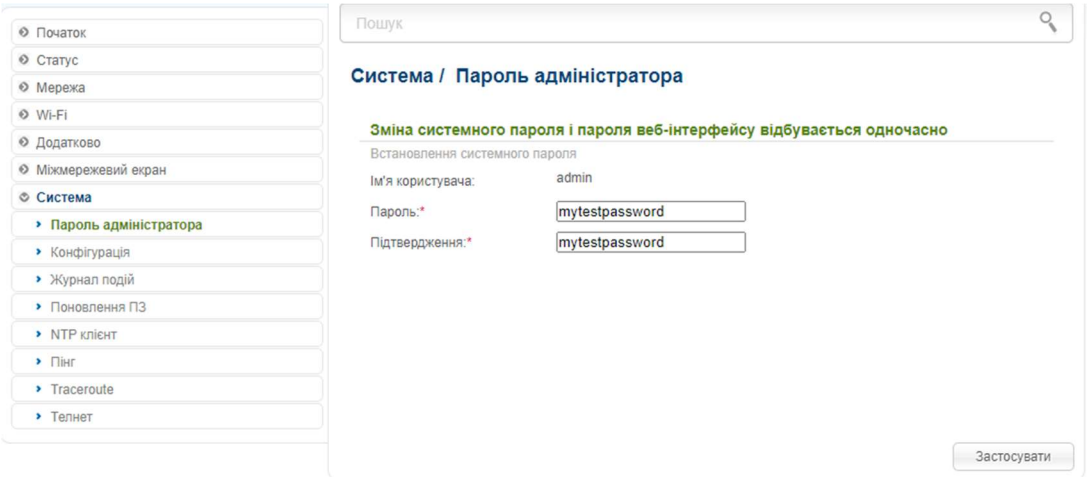

Рисунок 2.6 — встановлення паролю маршрутизатора

У словник із паролями помістимо декілька розповсюджених варіантів, а також рядок із встановленим нами паролем (рис. 2.7).

```
1   123123
2   654321
3   123456
4   qwerty123
5   qwerty
6
7   password
8   mytestpassword
9   easypassword
10  admin
```

Рисунок 2.7 — вміст файлу password.txt

Запустивши програму та ввівши потрібні дані, побачимо процес перебору у вигляді відображення паролю, який наразі перевіряється, коли черга дійде до потрібного рядку, вона сповістить нас про успішне завершення та покаже результат (рис. 2.8).

Так як ми будемо перевіряти наш роутер, то використовуємо його локальну адресу — 192.168.0.1, порт за замовчуванням — 23, логін також залишився незмінним — admin.

```
Enter ip-address: 192.168.0.1
Enter Telnet Port: 23
Enter Login: admin
123123

654321

123456

qwerty123

qwerty



password

Password found!
admin and mytestpassword


Process finished with exit code 0
```

Рисунок 2.8 — результат виконання перебору паролів

Бачимо, що програма дійсно знайшла потрібний пароль та зупинила своє виконання.

2.1.3 Анонімна автентифікація

Найчастіше сервіси, які мають доступ до конфігурацій маршрутизатора, так чи інакше, захищені логіном та паролем, тобто для отримання можливості налаштовувати роутер користувачу потрібні автентифікаційні дані. Проте існує вразливість, яка має назву «Anonymous Access», назва якої говорить сама про себе, вона дозволяє переглядати, або в деяких випадках навіть змінювати, параметри без необхідності входу, анонімно.

Через цю вразливість у 2017 із FTP серверів хакерами було викрадено персональну інформацію та інформацію про стан здоров'я клієнтів медичних та стоматологічних закладів [15]. Відповідно до списку [16], опублікованого Міністерством охорони здоров'я та соціальних служб США , у березні 2017 року було зламано більше 3,5 мільйонів медичних записів. Дані охорони здоров'я є найбільш бажаним типом даних для хакерів і кіберзлочинців, набагато більше, ніж дані кредитних карток.

Підприємства в усьому світі покладаються на протокол SSH для віддаленого адміністрування своїх комп'ютерних систем. Часто SSH використовується для передачі даних для виконання дій, які вимагають найвищих адміністративних привілеїв.

Різні програми SSH використовують деякі з найнадійніших доступних шифрів, наприклад AES128-CBC, Blowfish-CBC і AES256-CTR. Отже, припускаючи, що реалізація хороша, SSH досить важко зламати. Ось чому багатьом фахівцям з кібербезпеки подобається ця технологія.

Але не так давно виявилося, що Libssh версії 0.6 і вище має уразливість обходу автентифікації в коді сервера. Надаючи серверу повідомлення SSH2_MSG_USERAUTH_SUCCESS замість повідомлення SSH2_MSG_USERAUTH_REQUEST, яке сервер очікував би щоб ініціювати

автентифікацію, зловмисник міг успішно пройти автентифікацію без жодних облікових даних. Уразливість існує більше чотирьох років.

Виявив помилку Пітер Вінтер-Сміт з NCC Group. Як повідомляє Ars Technica Вінтер-Сміт сказав:

«Проблема в основному пов’язана з помилкою в бібліотеці libssh, яка є результатом того, що сервер використовує однакову систему для автентифікації і клієнтів, і серверів.

Код відправлення повідомлень, який обробляє повідомлення в режимі клієнта або в режимі сервера (одна і та ж функція), не гарантує, що тип отриманого повідомлення підходить для режиму, у якому воно працює. Тому, наприклад, сервер надсилатиме повідомлення, які розроблено лише для обробки на стороні клієнта, навіть якщо вони працюють у режимі сервера.

Повідомлення SSH2_MSG_USERAUTH_SUCCESS використовується сервером для інформування клієнта про успішну автентифікацію, воно оновлює внутрішній стан системи libssh, щоб позначити клієнта як автентифікованого на сервері. Я виявив, що якщо на сервер надсилається те саме повідомлення, він оновлює внутрішній стан системи, щоб повідомити серверу, що клієнт автентифікований».

2.1.4 Атака NAT slipstreaming

Атака NAT slipstreaming, представлена дослідником безпеки Самі Камкаром, дозволяє при відкритті сторінки в браузері створювати підключення сервера атакуючого із будь-яким портом UDP або TCP на системі користувача.

Суть атаки полягає в обмані механізму відслідковування підключень Application Level Gateways (ALG) у трансляторах адрес або міжмережевих екранах, який використовується для організації проброса через NAT протоколів, у яких декілька мережевих портів (один для даних, інший для керування), таких як SIP, H323, IRC DCC та FTP. Атака може бути використана до користувачів, які мають внутріншю адресу із інтранет

діапазону (192.168.x.x, 10.x.x.x) і дозволяє відправити на будь-який порт будь-які дані (без заголовків HTTP).

Для здійснення атаки достатньо, щоб жертва запустила підготовлений зловмисником JavaScript-код, наприклад, відкривши сторінку на сайті або подивилась шкідливу рекламну вставку на легітимному сайті.

Всього атака складається із трьох частин. На першому етапі атакуючий отримує дані про внутрішню адресу користувача за допомогою WebRTC або іншим доступним способом.

На другому етапі JavaScript-код, який виконується в браузері жертви, генерує великий (такий, який би не помістився в один пакет) HTTP POST запит на сервер атакуючого, використовуючи нестандартний номер мережевого порта для ініціювання налаштування параметрів сегментації TCP та розміру MTU в TCP-стеку жертви. У відповідь сервер атакуючого повертає TCP-пакет із опцією MSS (Maximum Segment Size що визначає максимальний розмір прийнятого пакета. У разі UDP маніпуляція схожа, але заснована на надсиланні великого запиту WebRTC TURN для виклику фрагментації на рівні IP.

Основна ідея в тому, що, знаючи параметри фрагментації, потім можна відправити великий HTTP-запит, хвіст якого припаде на другий пакет. При цьому хвіст, що потрапляє до другого пакету, підібрано так, що він не містить HTTP-заголовків і відсікається на даних, які повністю відповідають іншому протоколу, для якого підтримується обхід NAT.

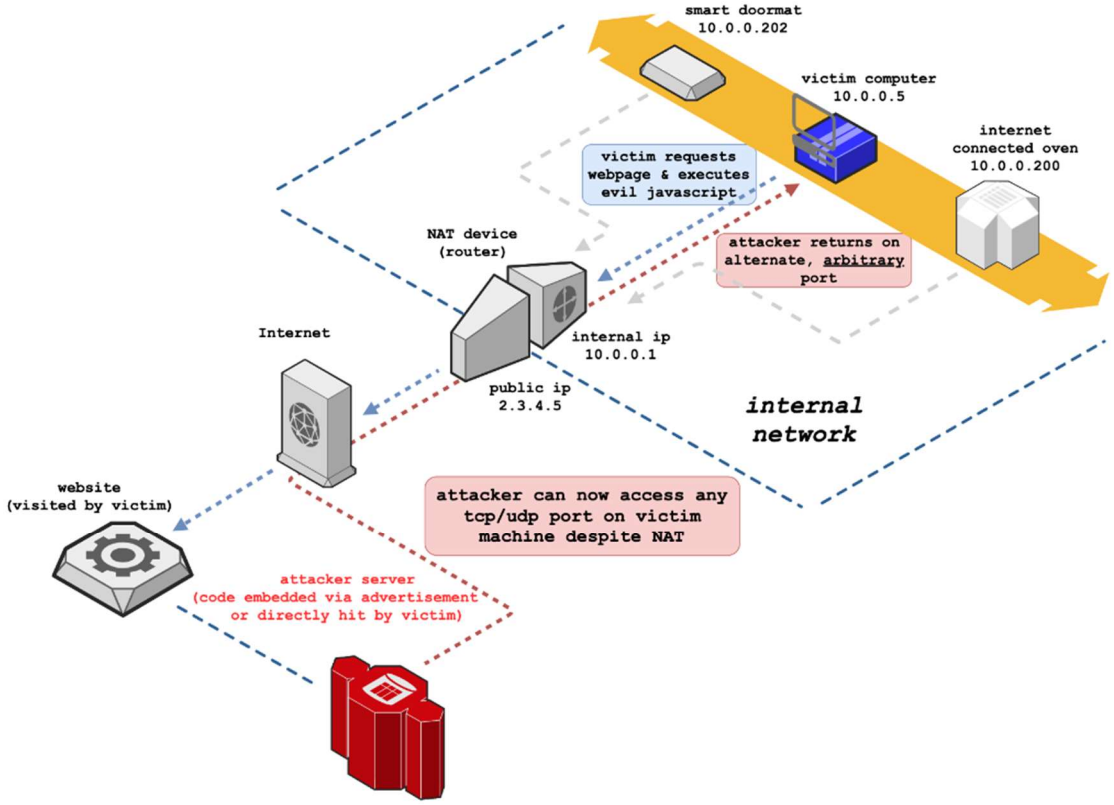

Рисунок 2.9 — Структурна схема здійснення атаки NAT slipstreaming [17]

На третьому етапі, за допомогою вищеописаної маніпуляції, JavaScript-код генерує і надсилає на TCP-порт 5060 сервера атакуючого спеціально підібраний HTTP-запит (або TURN для UDP), який після фрагментації розділиться на два пакети - пакет із HTTP-заголовками та частиною даних і коректний SIP-пакет, у якому зазначено внутрішній IP жертви. Система відстеження з'єднань у мережевому стеку вважатиме, що цей пакет є початком сеансу за протоколом SIP, і ввімкне перекидання пакетів для будь-якого вибраного атакувальником порту, вважаючи, що цей порт використовується для каналу передавання даних.

Атака може бути здійснена незалежно від використовуваного браузера. Для вирішення проблеми розробники Mozilla запропонували заблокувати можливість надсилання HTTP-запитів на мережеві порти 5060 і 5061, пов'язані

з протоколом SIP. Аналогічний захід захисту також мають намір реалізувати розробники Chromium, а також деякі інші розробники.

2.1.5 Комбінація DHCP starvation і Rogue DHCP Server

Спрощена схема роботи DHCP:

- **Discover:** клієнт, який не має IP-адреси, посилає широкомовний запит на адресу 255.255.255.255, у якому просить відповісти наявні в мережі DHCP-сервери.
- **Offer:** DHCP-сервери надсилають відповідь, у якій пропонують параметри конфігурації (IP-адреса, DNS-сервери, default gateway). Відповідь відсилається на MAC-адресу клієнта.
- **Request:** клієнт обирає, з яким сервером (за наявності декількох) йому зручніше працювати і надсилає запит адреси. Цей запит також надсилається широкомовно, але як одна з опцій уже вказується IP-адреса конкретного сервера.
- **Acknowledgment:** на цьому етапі запит підтверджується сервером. Після отримання цього пакета клієнт конфігурує свої мережеві параметри і процес отримання адреси можна вважати таким, що відбувся.

Мета Rogue DHCP Server атаки — підміна DHCP-сервера. При одночасному знаходженні в мережі двох DHCP-серверів, один з яких "ворожий", деяка частина клієнтів сконфігурує у себе неправильні адреси та інші мережеві реквізити.

Внаслідок підміни шлюзу за замовчуванням неавторизований DHCP-сервер отримає можливість прослуховувати весь трафік клієнтів, перенаправляючи надалі пакети за призначенням. Таким чином ми маємо найпростішу реалізацію атаки типу MitM (Man in the Middle), яка може бути здійснена в більшості сучасних мереж.

DHCP starvation — ще одна атака, здійснювана за допомогою протоколу DHCP. DHCP-пул, з якого клієнти отримують IP-адреси, обмежений.

Наприклад, це може бути 253 адреси (при масці 255.255.255.0). Для нормальної роботи мережі цього має вистачити всім клієнтам, але атака DHCP starvation прагне змінити цю ситуацію. Як відбувається атака:

1. Атакуючий пристрій запитує собі IP-адресу в DHCP-сервера й отримує її;
2. MAC-адреса атакуючого пристрою змінюється і він запитує наступну, вже іншу IP-адресу, маскуючись під нового клієнта;
3. Такі дії повторюються доти, доки весь пул IP-адрес на сервері не буде вичерпано.

Далі можливі два наслідки, залежно від того, що є метою атаки:

- Відмова в обслуговуванні. IP-адреси вичерпано, і нові хости не можуть отримати їх. Таким чином, їхня взаємодія з мережею на цьому закінчиться.
- Підміна DHCP-сервера. DHCP starvation чудово комбінується з попередньою атакою. Оскільки на основному DHCP-сервері вільних адрес не залишилося, він вибуває з гри, і 100% клієнтів дістається ворожому DHCP-серверу, що атакує.

2.2 Порти та їх призначення

2.2.1 Теоритичні відомості про мережеві порти

Порти — це логічні конструкції, які ідентифікують певний тип мережевої служби. Кожен порт пов'язаний з певним протоколом, програмою або службою і має номер порту для цілей ідентифікації. Наприклад, захищені повідомлення за протоколом передачі гіпертексту (HTTPS) завжди надходять на порт 443 на стороні сервера, порт 1194 призначений виключно для OpenVPN, а порт 20 використовується протоколом передачі данних File Transfer Protocol (FTP).

Найбільш поширеними транспортними протоколами є протокол керування передачею (Transmission Control Protocol, TCP) та протокол користувацьких дейтаграм (User Datagram Protocol, UDP). TCP — це протокол, орієнтований на встановлення з'єднання, з вбудованими засобами повторної передачі та відновлення помилок. UDP — це протокол без з'єднання, який не відновлює і не виправляє помилки в повідомленнях; він швидший і має менше мережевого трафіку, ніж TCP. І TCP, і UDP знаходяться на транспортному рівні стека TCP/IP і використовують протокол IP для адресації та маршрутизації даних в Інтернеті. Програмне забезпечення та послуги призначені для використання TCP або UDP, залежно від їхніх вимог.

Порти TCP і UDP можуть знаходитись в одному з трьох станів:

- **Open** — відповідає на запити на з'єднання.
- **Closed** — порт недоступний, що свідчить про відсутність відповідного сервісу.
- **Filtered** — брандмауер відстежує трафік і блокує певні запити на підключення до порту.

2.2.2 Ризики безпеки, пов’язані з портами

Численні інциденти продемонстрували, що відкриті порти найбільш вразливі до атак, коли служби, які їх прослуховують, неоновленні, недостатньо захищені чи неправильно налаштовані, що може призвести до скомпрометованих систем і мереж. У цих випадках зловмисники можуть використовувати відкриті порти для здійснення різних кібератак, які використовують відсутність механізмів автентифікації в протоколах TCP і UDP. Одним із поширених прикладів є спуфінг, коли зловмисник імітує систему чи службу та надсилає зловмисні пакети, часто в поєднанні з IP-спуфінгом і атаками типу "людина посередині". Крім того, навмисно відкриті порти (наприклад, на веб-сервері) можуть бути атаковані за допомогою атак на прикладному рівні, таких як SQL ін’єкція.

Іншим поширеним прийомом є атака на відмову в обслуговуванні (DoS), яка найчастіше використовується у формі розподіленої відмови в обслуговуванні (DDoS), коли зловмисники надсилають величезну кількість запитів на підключення від різних машин до служби на цільовій службі, щоб вичерпати її ресурси.

2.2.3 Вразливі порти, які потребують вашої уваги

Будь-який порт може стати мішенню зловмисників, але деякі з них з більшою ймовірністю стануть жертвами кібератак, оскільки вони зазвичай мають серйозні недоліки, такі як уразливість додатків, відсутність двофакторної автентифікації та слабкі ідентифікаційні дані.

Ось найбільш вразливі порти, які регулярно використовуються в атаках:

Таблиця 2.2 — призначення найбільш вразливих портів

Номер порта	**Найменування служби**	**Призначення**
20, 21	File Transfer Protocol	Передача даних
22	Secure Shell	Забезпечення доступу до серверів
23	Telnet	З’єднання із віддаленим комп’ютером
25	Simple Mail Transfer Protocol	Робота із електронними листами
53	Domain Name Service	Система доменних імен
137, 139, 445	NetBIOS та System Message Block	Робта із серверами
80, 443, 8080, 8443	HTTP і HTTPS	Протокол передачі даних
1433, 1434, 3306	Databases	Робота із базами даних
3389	Remote Desktop	Робота із віддаленим ПК

Ports 20 and 21 (FTP). FTP відомий як застарілий і небезпечний. Тому зловмисники часто використовують його в своїх цілях за допомогою:

- Підбору паролів
- Анонімної автентифікації (можна увійти в порт FTP за допомогою імені користувача та пароля «anonymous»)
- Міжсайтового скриптингу
- Атаки через каталог

Port 22 (SSH). Порт 22 призначений для Secure Shell (SSH). Це TCP-порт для забезпечення безпечного доступу до серверів. Хакери можуть використовувати порт 22 за допомогою витоку SSH-ключів або підбору облікових даних методом брутфорсу.

Port 23 (Telnet). Порт 23 — це протокол TCP, який з'єднує користувачів із віддаленими комп'ютерами. Здебільшого Telnet було замінено SSH, але деякі веб-сайти все ще використовують його. Оскільки він застарілий і небезпечний, він вразливий до багатьох атак, зокрема підбору облікових даних, підробки та перехоплення облікових даних.

Порт 25 (SMTP). Порт 25 — це порт SMTP (Simple Mail Transfer Protocol) для отримання та надсилання електронних листів. Без належної конфігурації та захисту цей TCP-порт вразливий до спуфінгу та спаму.

Порт 53 (DNS). Порт 53 призначений для системи доменних імен (DNS). Це порт UDP і TCP для запитів і передачі відповідно. Цей порт особливо вразливий до DDoS-атак.

Порти 137 і 139 (NetBIOS через TCP) і 445 (SMB). Блок повідомлень сервера (SMB) використовує порт 445 безпосередньо та порти 137 і 139 опосередковано. Кіберзлочинці можуть використовувати ці порти за допомогою:

- Використання експлойта EternalBlue , який використовує вразливості SMBv1 у старих версіях комп'ютерів Microsoft (хакери

використовували EternalBlue на порту SMB для поширення програми-вимагача WannaCry у 2017 році).

- Захоплення хешів NTLM.
- Підбір облікових даних SMB для входу.

Порти 80, 443, 8080 і 8443 (HTTP і HTTPS). HTTP і HTTPS є найпопулярнішими протоколами в Інтернеті, тому вони часто стають мішенню зловмисників. Вони особливо вразливі до міжсайтових сценаріїв, ін'єкцій SQL, підбору міжсайтових запитів і DDoS-атак.

Порти 1433, 1434 і 3306 (використовуються базами даних). Це стандартні порти для SQL Server і MySQL. Вони використовуються для розповсюдження шкідливого програмного забезпечення або піддаються прямим атакам у сценаріях DDoS. Досить часто зловмисники перевіряють ці порти, щоб знайти незахищену базу даних із конфігураціями за замовчуванням, які можна використовувати.

Порт 3389 (віддалений робочий стіл). Цей порт використовується в поєднанні з різними вразливими місцями в протоколах віддаленого робочого столу та для перевірки витоку або слабкої автентифікації користувача. Уразливості віддаленого робочого столу наразі є найбільш використовуваним типом атак; одним із прикладів є вразливість BlueKeep.

2.2.4 Дослідження Alert Logic уразливості портів

У 2019 році компанія Alert Logic провела дослідження [18], яке базувалося на ретельному вивченні понад 1,3 петабайт даних про безпеку, понад 2,8 мільярдів IDS-подій, 8,2 мільйона підтвердженних інцидентів та типових вразливостей, присутніх у малому та середньому бізнесі.

У своєму звіті компанія Alert Logic, перерахувала основні недоліки у системах захисту, які спостерігаються під час атаках на понад 4000 її клієнтів.

Відповідно до звіту, порти, які найчастіше використовуються для здійснення атак, — це 22, 80 і 443, які відповідають SSH (Secure Shell), HTTP

(Hypertext Transfer Protocol) і HTTPS (Hypertext Transfer Protocol Secure) (рис. 2.10).

Alert Logic каже, що вони з'являються в 65% інцидентів, і це має сенс, оскільки вони мають бути відкритими для спілкування, будь то захищеним чи простим текстом.

На четвертому місці – порт для протоколу віддаленого робочого столу Microsoft (RDP), який відповідає за віддалений зв'язок між машинами. Цього року RDP привернув увагу завдяки численним виправленням уразливостей, які призводять до віддаленого виконання коду

Порти, позначені як серйозний ризик, призначені для протоколу передачі файлів (FTP - 20, 21). Активні сервери були виявлені на принтерах, камерах і джерелах безперебійного живлення, які, за оцінками, становлять до третини всіх виявлених FTP-серверів.

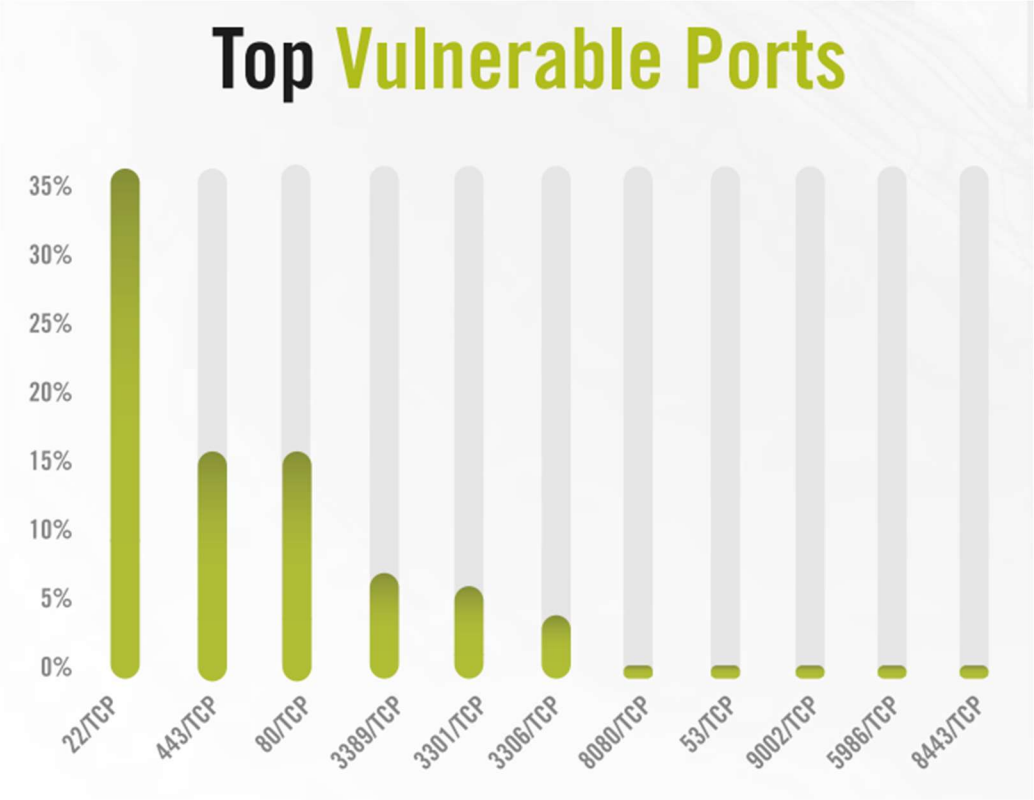

Рисунок 2.10 — порти, які частіше всього стають мішенню кібератак [18]

2.3 Сценарії протидії злому

План дій у разі фіксування спроби отримати несанкціонований доступ до мережі, яку ми хочемо захистити в найкращому випадку повинен включати в себе:

- Попередження
- Протоколювання
- Передача сформованного протоколу (IP, порти, час)
- Провокація на підбор фіктивного паролю
- Провокація на передачу троянської програми на фіктивний диск

У якості попередження буде використано звичайний текстовий напис на дисплеї пристрою, який повинен містити IP-адресу машини, з якої надійшов запит, а також порт або порти, куди ці запити було надіслано.

Важливо зафіксувати дані про напад, тобто потрібно створити спеціальний лог, де будуть зберігатися записи із IP-адресою зловмисника, порти, які атакуються і час, коли відбувалися атаки. Аналізуючі ці дані можна приймати відповідні дії щодо захисту: блокування IP-адреси, закриття портів, які частіше всього виступають мішенню, приділяти більшу увагу у конкретні проміжки часу і тд.

У найкращому сценарії ми повинні передвати сформований протокол із даними про кібератаки на зовнішній комп’ютер, так ми матимемо змогу розпоряджатися зібраними даними, як нам завгодно. І в такому випадку наш прилад буде виступати не кінцевим пристроєм, а лише проміжною ланкою.

3 ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ

3.1 Вимоги до програмно-апаратного комплексу. Функціонування

Програмно-апаратний комплекс повинен представляти собою спеціальний прилад, який буде працювати як окрема захисна ділянка на яку, з маршрутизатора, будуть передаватись пакети. Візуально структурну схему можна зобразити так:

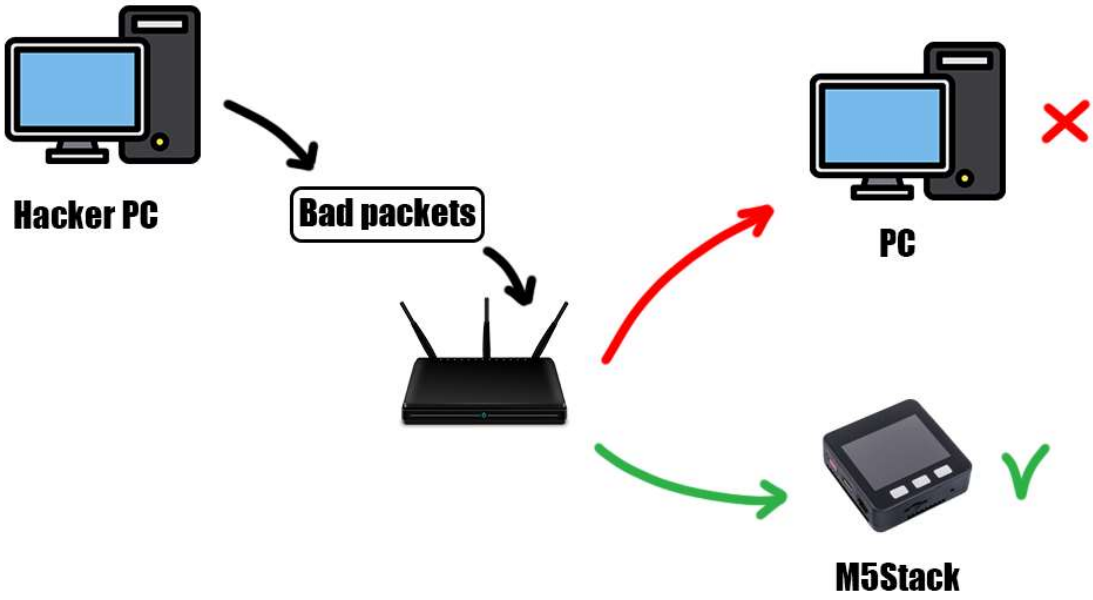

Рисунок 3.1 — структурна схема функціонування приладу

Комплекс повинен своєчасно реагувати на можливі загрози внутрінішній мережі, а саме: фіксувати такі спроби зловмисників як сканування портів, підбір паролю, спроби завантаження шкідливиго програмного забезпечення (таких як троянська програма, програми для встановлення бекдору та ін.) Після фіксації будь-якого небажаного сценарію комплекс має сповістити користувача, відобразивши на дисплеї сповіщення із інформацією про те, що відбувається спроба злому.

Для того, щоб відловлювати акти злому, які представляють із себе відправку пакетів, потрібно переадресовувати їх із роутера, на який вони надходять, на наш пристрій. Зробити це можна за допомогою, так званого, пробросу портів. Прокидання або перенаправлення порту — це операція, що входить до комплексу функцій перетворення мережевих адрес або NAT (механізм NAT передбачає використання одного зовнішнього інтерфейсу кількома пристроями локальної мережі). Прокидання передбачає зіставлення заданого порту на зовнішньому інтерфейсі роутера з певним портом пристрою, що запитується, у локальній мережі.

Також має бути створений та поміщений в лог новий запис, де буде вказано IP-адресу, порт, який перебуває під загрозою, і час. У найкращому випадку лог повинен автоматично, при досягненні певної кількості записів, передаватися на зовнішній комп’ютер через Wi-Fi.

3.2 Вимоги до апаратної частини

Із дослідження випливає, що у якості апаратної частини повинен бути прилад, який має змогу підключатися до внутрішньої мережі (роутера) за допомогою Wi-Fi.

Також пристрій повинен аналізувати пакети, які будуть переадресовуватися на нього із маршрутизатора, тобто для цього повинні існувати спеціалізовані бібліотеки, які дозволять приймати пакети, оброблювати їх, а також дозволяють передвавати їх на зовнішній комп’ютер.

Прилад повинен мати зовнішній дисплей для відображення повідомлення у разі фіксування атаки, розміри екрану не повинні бути надто громіздкими, 1 - 2 дюйми буде оптимальним варіантом, розширення можна обирати будь-яке, підійде навіть мінімальне 120 на 160.

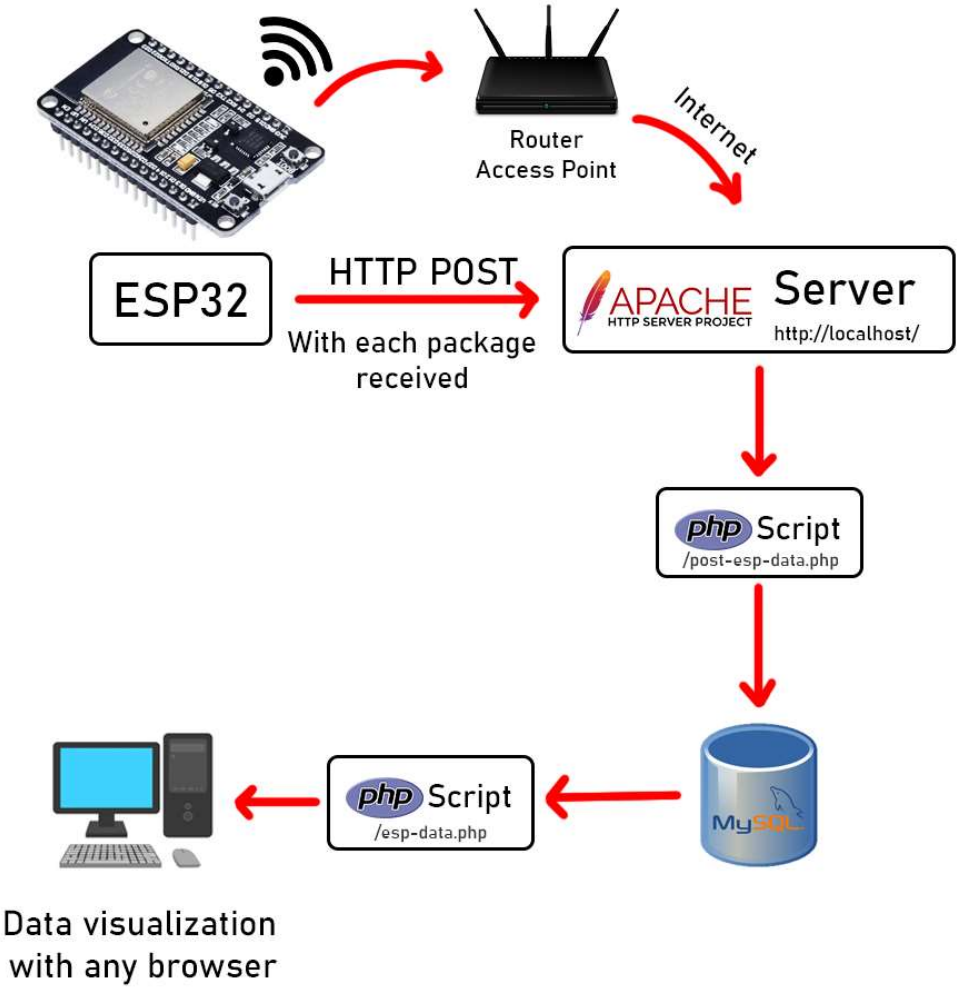

Рисунок 3.2 — структурна схема програмної частини

Для того, щоб зберігати логи, чудовою ідеєю є використання віддаленої бази даних. Цей варіант звільняє нас від потреби використовувати локальну пам’ять приладу, а також це набагато зручніше, ніж зберігання даних в простому текстовому файлі, так як їх обробка та аналіз значно поліпшиться.

4 РОЗРОБКА

4.1 Апаратна частина

По суті, у якості апаратної частини для обраної цілі можна було б створити свій прилад на будь-якій платі по типу ESP8266, Arduino Uno, Raspberry Pi Zero, Nodemcu, Beaglebone і так далі. Кожна із них має свої переваги та недоліки, до кожного із цих контролерів можна було б докупити та під'єднати дисплей, написати свою конфігурацію та підлаштувати під власні цілі. Але слід звернути увагу, що далеко не всі контролери "із коробки" мають підтримку Wi-Fi, що є дуже важливим у даному випадку, також треба розуміти, що контролери постачаються без будь-яких корпусів, їх доведеться розроблювати власноруч.

Далі наведено коротку характеристику найпопулярніших варіантів плат, їх плюси і мінуси.

4.1.1 Arduino Uno

Arduino Uno — мікроконтролерна плата на основі **ATmega328P**. Вона має 14 цифрових входів/виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, керамічний резонатор 16 МГц (CSTCE16M0V53-R0), USB-роз'єм, роз'єм живлення, роз'єм ICSP і кнопку скидання.

Станом на 2022 рік на офіційному сайті Arduino Uno коштує 27,60 долларів, тобто приблизно 1000 гривень.

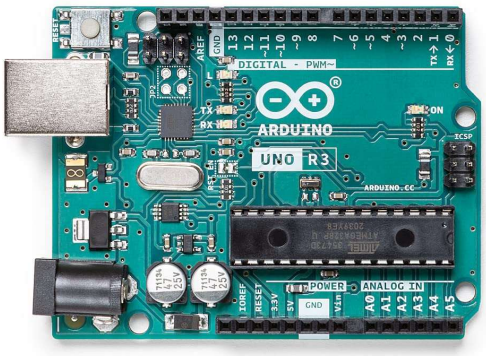

Рисунок 4.1 — Arduino Uno

Із плюсів хотілось би відмітити:

- драйвер USB-to-serial дозволяє підключити Arduino безпосередньо до будь-якого комп’ютера та почати програмування. Порти вводу/виводу забезпечують достатньо швидке створення прототипів і усунення несправностей.
- Arduino Uno є самодостатнім. Все, що йому потрібно для функціонування, це живлення і код. Якщо у вас є досвід роботи зі схемами, ви можете працювати напряму з чіпом ATmega328 або з будь-яким іншим. Однак для роботи з Arduino Uno знання схем не потрібні.
- Є безліч докладних і добре прокоментованих прикладів коду, вбудованих у надану Arduino IDE. Ці ресурси також можна знайти на домашній сторінці Arduino.

Із мінусів:

- Без додавання макетної плати побудувати якусь складну схему буде важко.
- Не надто висока потужність, всього 32 кБ флеш пам’ті та 2 кБ ПЗУ
- Можливо, це моя суб’єктивна точка зору, але хотілось би бачити USB-C.

4.1.2 Raspberry Pi Zero

Це найтонший і максимально урізаний Raspberry Pi на сьогодні. Цей мікрокомп'ютер може запустити повноцінний дистрибутив Linux.

У комплекті він має гніздо для карти micro SD, порт mini HDMI, два порти micro USB (один для живлення, один для USB) і 512 МБ оперативної пам’яті та одноядерний процесор 1 ГГц.

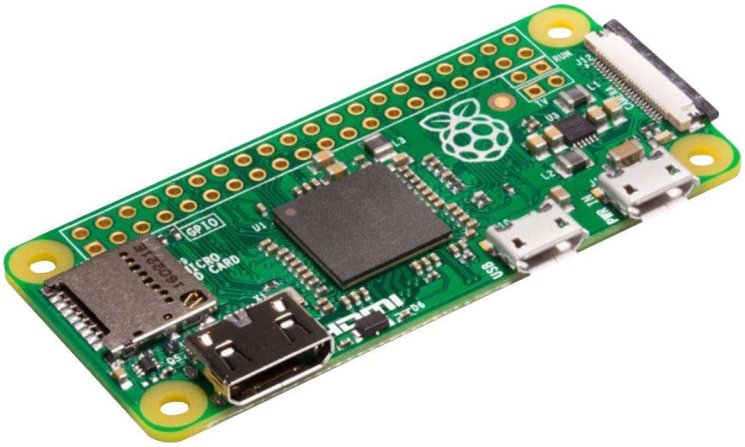

Рисунок 4.2 — Raspberry Pi Zero

Його характеристик навряд чи вистачить на серйозний проєкт, також немало важливим мінусом є відсунтність WiFi/Bluetooth, а також потреба докупляти аксесуари. Коштує плата всього 5-8 долларів.

4.1.3 Nodemcu

NodeMCU – це доволі цікавий мікроконтролер, який пропонує безліч можливостей, особливо з огляду на його ціну, яка складає 3 доллари за найдешевшу версію. На відміну від Arduino, який використовує 8 бітний ATMEA з частотою 16 МГц, тут використовується чіпсет ESP8266 з 32 бітним процесором Tensilica Xtensa LX106 80 МГц, також тут є WiFi, вбудована підтримка TCP/IP, 4 мегабайти вбудованого сховища. Крім цього тут є 10 виходів GPIO.

До пристрою можна підключати різні компоненти, такі як монітори, сенсори або сервоприводи.

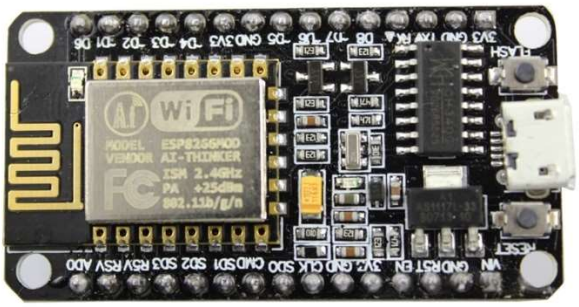

Рисунок 4.3 — NodeMCU

Як і Arduino, його дуже просто використовувати, достатньо написати код і завантажити його на пристрій USB. Програми пишуться на Lua, це інтерпретована мова програмування, схожа на Python або Ruby.

4.1.4 Beaglebone

Beaglebone використовує ARM Cortex V8 із частотою 700 МГц та 256 мегабайт оперативної пам'яті DDR2, а також флешку об'ємом 4 Гб. Як мови програмування можуть використовуватися Python, C, C++, PHP, JavaScript. Пристрій підтримує встановлення SD картки, а також є USB порт, через який можна підключати різні розширення, наприклад Ethernet або інший комп'ютер. Ціна варіюється від 60 до 80 долларів.

Із особливостей виділяють:

- 512 МБ DDR3 DRAM
- Вбудовану флеш-пам'ять eMMC

- Продуктивність 1 ГГц
- Підтримка Ubuntu і Android
- uHDMI, uSD, USB, Ethernet, 5 В
- Сумісність із більшістю існуючих додаткових плат Cape
- Вбудована пам'ять 2 ГБ з Angstrom

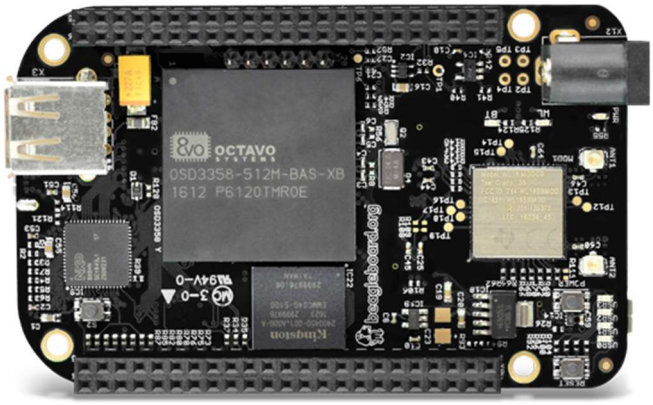

Рисунок 4.4 — Beaglebone

4.1.5 ESP8266

ESP8266 – це мікроконтролер з підтримкою Wifi, який набирає популярності серед ентузіастів в електроніці. Він коштує дешевше за Arduino, але може програмуватися в тій же Arduino IDE. Але для його живлення потрібно подавати 3.3 вольта, а не 5. Пристрій поставляється з вбудованим регулятором живлення та кількома портами вводу-виводу.

Враховуючи низьку вартість, невеликий розмір і можливість адаптації до вбудованих пристроїв, ESP8266 зараз широко використовується на пристроях Інтернету речей. Незважаючи на те, що на зміну йому прийшло чіп мікроконтролера ESP32 нового покоління, ESP8266 все ще залишається популярним вибором для розробників і виробників IoT.

Модуль ESP8266 дозволяє підключатися до Wi-Fi 2,4 ГГц за стандартом IEEE 802.11. Його можна використовувати з вбудованим програмним забезпеченням ESP-AT для підключення Wi-Fi до зовнішніх хостів MCU, або його можна використовувати як самодостатній MCU, запустивши SDK на основі RTOS. Модуль має повний стек TCP/IP і забезпечує можливість обробки даних, читання та керування GPIO (General-purpose input/output).

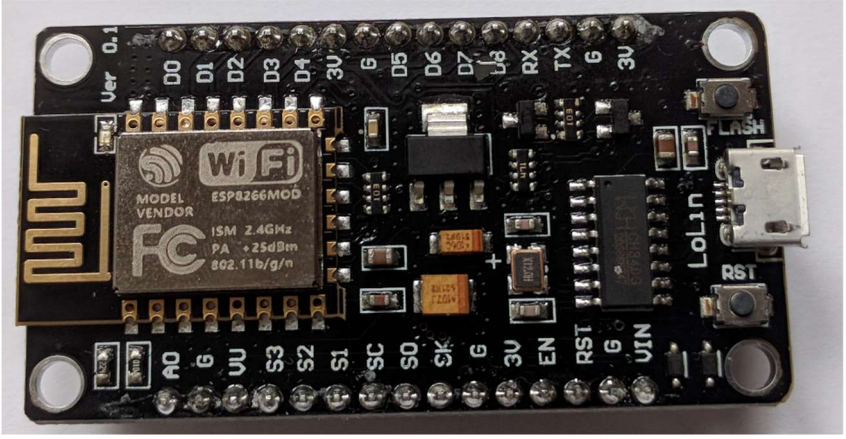

Рисунок 4.5 — ESP8266

ESP8266 має багато застосувань, коли мова йде про IoT. Ось лише деякі функції, для яких використовується мікросхема:

- **Мережа:** антена Wi-Fi модуля дозволяє вбудованим пристроям підключатися до маршрутизаторів і передавати дані.
- **Обробка даних:** включає обробку основних вхідних даних від аналогових і цифрових датчиків для набагато складніших обчислень за допомогою RTOS або SDK, що не входить до ОС.
- **Підключення P2P:** Можливість створювати прямий зв’язок між ESP та іншими пристроями за допомогою підключення IoT P2P.
- **Веб-сервер:** доступ до сторінок, написаних HTML або іншими мовами розробки.

4.1.1 Опис структури M5Stack

M5Stack — це технологічна компанія, яка розробляє та виробляє набір інструментів розробки з відкритим кодом, включаючи апаратне забезпечення, платформу програмування та рішення IoT. Вона був заснована Джиммі Лаєм у 2017 році та базується в Шеньчжені, Китай.

На сьогодні M5Stack представило безліч девайсів, за допомогою яких можна створити власний пристрій для будь-яких цілей. В основі всіх проєктів стоїть контролер, який може бути представлений одним із трьох варіантів: M5Stack, M5Stick та M5Atom. Всі вони виконані на основі мікроконтролера ESP32. Відрізняються ж вони розміром корпусу, наявністю екрана, ємністю батареї та продуктивністю мікроконтролеру, а також сумісністю із різного типу модулів, які розробляються спеціально під них.

В цілому M5Stack — це невеликий комп'ютерний пристрій, котрий містить в собі необхідні функції для розробки прототипу, такі як бездротовий зв'язок, дисплей, а також кнопки керування.

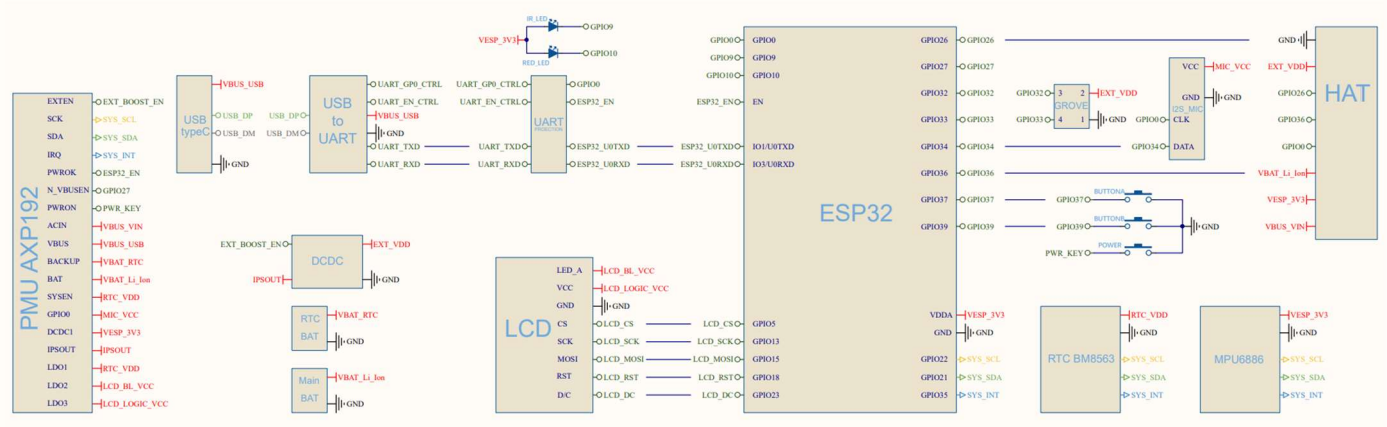

Рисунок 4.6 — принципова схема M5Stack

Станом на 2022 рік існує дві моделі M5Stack, перша, більш стара версія Basic Core, глобально нічим не відрізняється, крім дизайну, відустністю сенсорних кнопок та процесором (в старшій версії це Tensilika LX6 600 DMIPS).

Остання ж модель даного девайсу називається M5Core2. Працює він на мікроконтролері ESP32 D0WDQ6-V3, а також має двоядерний 32-розрядний процесор Xtensa® 240 МГц LX6, яким можна керувати окремо. В стандартній комплектації підтримується Wi-Fi та включено вбудовану флеш-пам’ять 16 МБ та 8 МБ PSRAM, інтерфейс USB TYPE-C для зарядки, завантаження програм і послідовного зв’язку, 2,0-дюймовий вбудований сенсорний екран і вбудований вібраційний двигун.

M5Core2 також має вбудований модуль RTC, який може забезпечити точний час. Джерелом живлення керує мікросхема керування живленням AXP192, яка може ефективно контролювати споживання електроенергії базою, а вбудований зелений світлодіодний індикатор живлення допомагає сповіщати користувача про рівень заряду батареї. Ємність акумулятора була підвищена до 390 мАг, що може живити ядро набагато довше, ніж у попередній моделі.

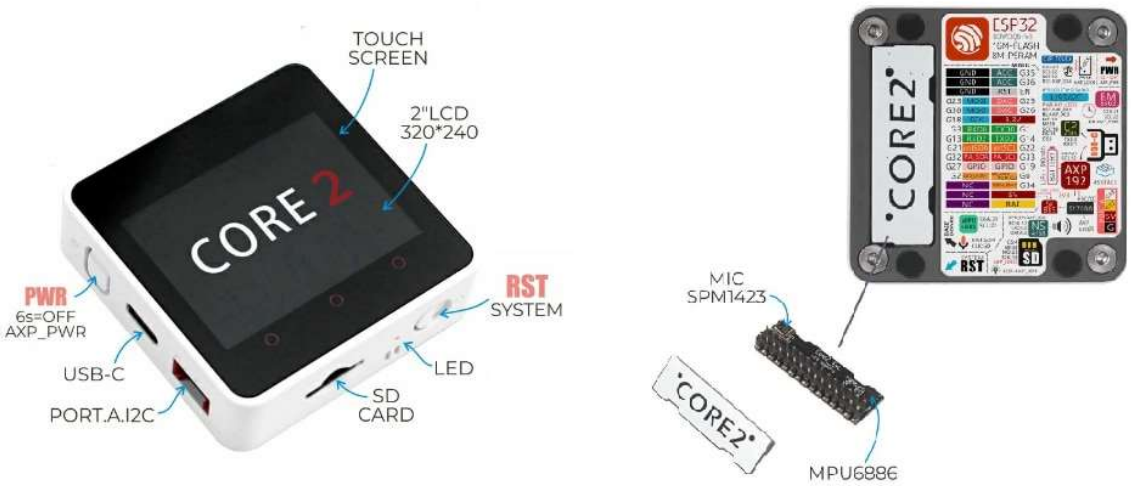

Рисунок 4.7 — M5Core2

M5Stack має слот для TF-карти (microSD) і динаміки. Для забезпечення більш високої якості звуку на виході використовується мікросхема підсилювача потужності цифрового аудіоінтерфейсу I2S, яка ефективно

запобігає спотворенню сигналу. На лівій стороні та внизу основи розташовані незалежні кнопки ввімкнення та скидання.

Три іконки на передній панелі екрану — це сенсорні кнопки, які можна програмувати. На задній панелі є невелика плата розширення з 6-осьовим датчиком IMU та мікрофоном. Платформи розробки та мови програмування, які підтримується M5Stack Core2: Arduino, UIFlow (з використанням мови Blockly, MicroPython).

4.1.1 Підсумок

Не так давно габарити комп’ютерів могли сягати розмірів кімнати, але із часом вони ставали все меншими і компактнішими, зараз вони з легкістю можуть поміщатися на столі, або навіть у вашому портфелі, якщо мова йде про ноутбуки. Прогрес рухається далі і сьогодні досить продуктивна машина вже може поміститися у вашій кишені. Мова йде про мікрокомп'ютери, такі як Arduino та Raspberry Pi. Вони дешеві, мають мінімальний розмір але їх потужності вже достатньо для багатьох цікавих та корисних проєктів, які можна використовувати у реальному житті.

Вище я порівняв характеристики таких популярних варіантів, як Arduino Uno, Raspberry Pi, NodeMCU, Beaglebone та ESP8266, кожен із них безперечно має свої переваги перед іншими та можуть стати оптимальним варіантом для вас. Для себе я знайшов деяку альтернативу у вигляді цілої платформи на основі із ESP32 — M5Stack, яка відразу постачається із дисплеєм, кнопками управління, спікерами, та багато чим іншим, а також все це упаковано в гарний та ергономічний корпус, а ціна не перевищує 2 тисяч гривень, що насправді є дуже привабливим показником. В доповнення до цього всього хмарні оновлення, аксесуари, бібліотеки, програмні пакети під будь-яку задачу. На мою думку, ніхто не наблизився до цього на ESP32.

4.2 Програмна частина

4.2.1 Налагодження роутера / шлюза у зовнішню мережу

Перед тим, як приступати до конфігурування роутера, потрібно розібратися із поняттям «проброс портів» (або ***port forwarding***) та «трансляція мережевих адрес».

Трансляція мережевих адрес (**NAT**, Network Address Translation) — механізм, який дозволяє всім пристроям внутрішньої мережі використовувати єдину IP-адресу зовнішнього інтерфейсу, через який відбувається підключення до Інтернету або зовнішньої мережі. Найросповсюдженіший випадок, коли на роутер виділяється одна публічна IP-адреса, через яку і здійснюється вихід до зовнішньої мережі, за якою працюють та отримують доступ пристрої домашньої мережі із локальними IP-адресами.

При найпростішому випадку при роботі NAT відбувається модифікація адреси джерела та назначення в мережевих пакетах (рис. 4.7). У пакетів, які приходять із зовнішньої мережі, змінюється адреса отримувача, а у пакетів із внутрішньої мережі — відправника.

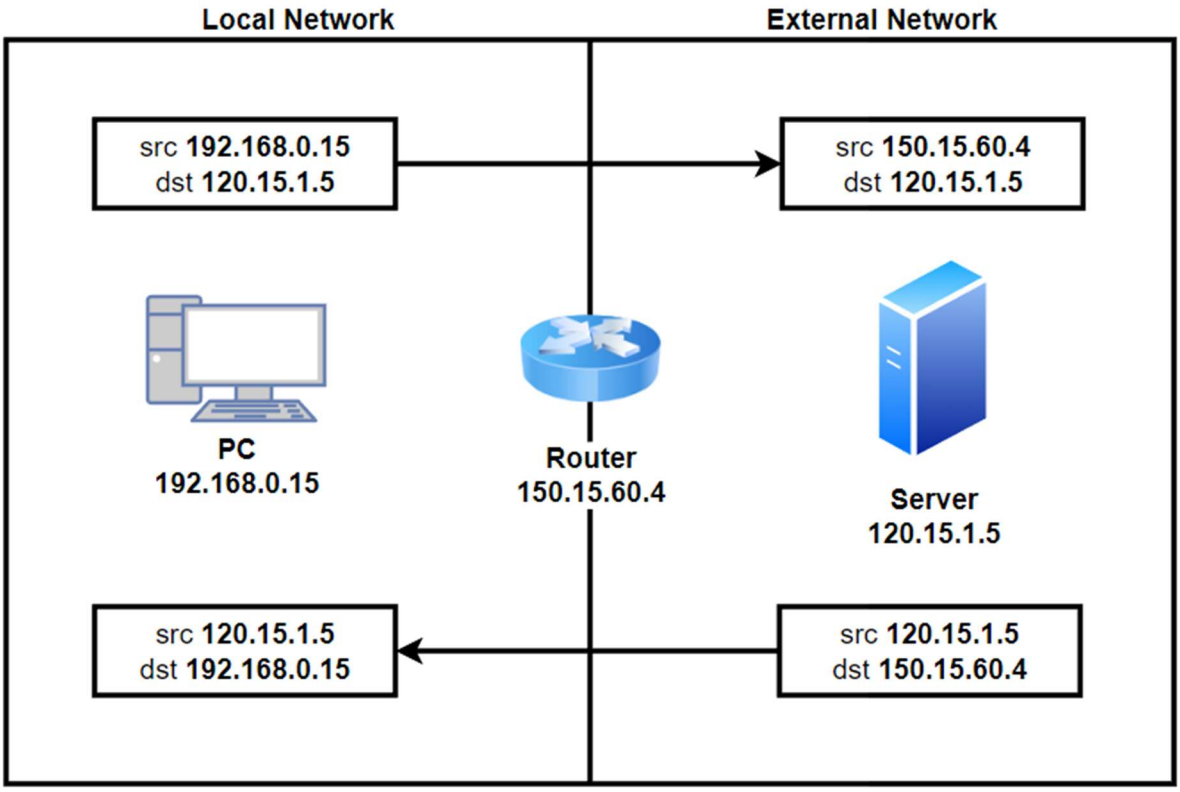

Рисунок 4.7 — підміна IP-адрес при роботі NAT

Простіше кажучи, NAT змінює IP-адресу джерела (внутрішню локальну/приватну адресу) в мережевому пакеті, прийнятому від пристрою локальної мережі, на глобальну/зовнішню адресу перед передаванням пакета в зовнішню мережу. При отриманні відповіді NAT перетворює адресу одержувача (зовнішню адресу) назад у локальну/внутрішню адресу перед передачею її вихідному внутрішньому хосту мережі.

За замовчуванням механізм NAT налаштований таким чином, щоб запобігати або обмежувати звернення ззовні (з боку зовнішньої мережі) до пристроїв локальної мережі, залишаючи можливість звернення з локальної мережі в зовнішню. NAT дає змогу приховати внутрішні сервіси комп'ютерів/серверів локальної мережі від звернень з Інтернету.

За допомогою користувацьких правил переадресації портів можна певні внутрішні сервіси (наприклад, Веб- або FTP-сервер), розташовані в локальній мережі за NAT, зробити видимими (доступними) для зовнішніх користувачів з Інтернету. Для цього потрібно створити правила NAT для трансляції (іноді

кажуть - для "перенаправлення", "прокидання", "відкриття") певних портів через роутер на комп'ютер/сервер локальної мережі (цей механізм також називають **Port Forwarding**). По суті, такі правила визначають трансляцію трафіку із зовнішньої мережі у внутрішню (такий тип правил NAT називають Destination NAT).

Отже, для того, щоб налагодити роутер для наших цілей, ми повинні налаштувати проброс портів. Зробити це можна в налаштувальній панелі роутера. Щоб потрапити туди, потрібно ввести локальну адресу вашого роутера в браузері, зазвичай це **192.168.0.1** або **192.168.1.1**.

Налаштування перенаправлення портів для мого роутера знаходяться в розділі «Міжмережевий екран» — «Віртуальні сервери» (рис. 4.8).

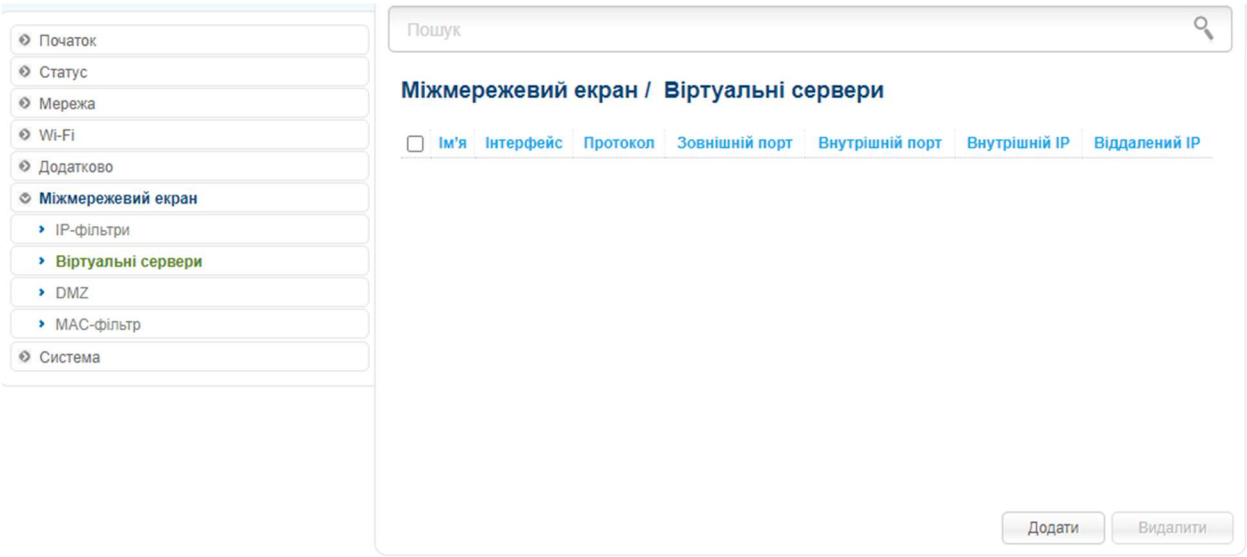

Рисунок 4.8 — Вікно налаштування перенаправлення портів на роутері D-Link

Після натискання кнопки «Додати» ми можемо вказати діапазон портів, які хочемо перенаправти, протокол пакетів, які хочемо перенаправляти (TCP/UDP), порт назначення, тобто куди пакети будуть перенаправлені, а також IP-адресу назначення (крім локальної адреси, також можна вказати віддалену) (рис. 4.9)

Рисунок 4.9 — Створення правила переадресування портів

Ми повинні створити правила для портів, які вважаємо найбільш вразливими (20, 21, 22, 53, 137, 80, 8080 і тд). У якості протокола будемо вказувати TCP/UDP, внутрішня адреса, на яку будуть переадресовуватися пакети – IP-адреса нашого пристрою на основі M5Stack.

Також, у рамках даної роботи важливо розібратися яким чином M5Stack буде співпрацювати з бездротовою мережею. Для цієї цілі існує спеціальна бібліотека, тобто набір спеціалізованих функцій для вирішення схожих за тематикою задач, яка має назву WiFi.h.

Ця бібліотека дає можливість створювати екземпляри серверів, клієнтів і надсилати/отримувати пакети UDP через WiFi. Можна підключатися до відкритих або зашифрованих мереж (WEP, WPA). IP-адресу можна призначити статично або через DHCP. Бібліотека також може керувати DNS.

Для того, щоб підключити M5Stack до бездротової мережі, створимо окрему функцію ***initWiFi()***, попередньо створивши константи із назвою точки доступу (***ssid***) та паролем (***password***) для неї.

```
void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  M5.Lcd.print("Connecting to WiFi..");
  while (WiFi.status() != WL_CONNECTED) {
    M5.Lcd.print('.');
    delay(1000);
  }
  M5.Lcd.setCursor(0, 0);
  M5.Lcd.clearDisplay();
  M5.Lcd.println("My local IP:");
  M5.Lcd.println(WiFi.localIP());
  M5.Lcd.println();
}
```

У результаті спочатку побачимо на дисплеї напис «Connecting to WiFi» і у разі успішного підключення до мережі, буде виведено локальну IP-адресу пристрою.

4.2.2 Розробка частини програми сповіщення про атаки

Як уже йшлося раніше, перед самим актом злому внітрішньої мережі, зловмисники повинні дізнатися, що саме вони будуть атакувати, які експлойти вони можуть використати. Ці дані вони отримують у результаті сканування портів, про це йшлося в пункті **2.1.1** даної роботи. Будь-який із способів сканування припускає, що зловмисник буде перебирати порти, передаючи на нашу IP-адресу деякі пакети.

Для початку розберемося що таке пакет і з чього він складається. Сьогодні існує два основних інтернет-протоколи, за допомогою яких передаються всі пакети, це **TCP** та **UDP**.

Протокол **TCP** (Transmission Control Protocol) — це мережевий протокол, який оптимізований під з'єднання. Інакше кажучи, перш ніж почати обмін даними, цьому протоколу потрібно встановити з'єднання між двома хостами. Цей протокол має високу надійність, оскільки дає змогу не втрачати дані під час передання, запитує підтвердження про одержання від сторони, що приймає, і за потреби відправляє дані повторно. При цьому пакети даних, що відправляються, зберігають порядок відправлення, тобто можна сказати, що

передача даних упорядкована. Мінусом цього протоколу є відносно низька швидкість передавання даних, через те, що виконання надійного та впорядкованого передавання займає більше часу, ніж в альтернативному протоколі UDP.

Протокол **UDP** (User Datagram Protocol), у свою чергу, простіший. Для передачі даних йому не обов'язково встановлювати з'єднання між відправником і одержувачем. Інформація передається без попередньої перевірки готовності сторони, що приймає. Це робить протокол менш надійним — під час передачі деякі фрагменти даних можуть губитися. Крім того, впорядкованість даних не дотримується — можливе непослідовне приймання даних одержувачем. Зате швидкість передачі даних за цим транспортним протоколом буде вищою.

Через таку відмінність протоколів, хедери пакетів також різняться вмістом між собою (рис. 4.10, 4.11). Дані, які потрібні нам, в рамках проєкту, від пакетів — це порт призначення (Destination port) та IP-адреса відправника (Source IP address). Порт ми зможемо дізнатися із TCP/UDP сегменту, а IP адреса міститься в IP хедері.

UDP Datagram Header Format

Bit #	0 — 7	8 — 15	16 — 23	24 — 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

Рисунок 4.10 — вміст UDP хедеру

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

Рисунок 4.11 — вміст TCP хедеру

Методика обробка пакетів даних протоколів також має свої відмінності, в рамках даної роботи було використано бібліотеку **WiFi.h**, для роботи із TCP пакетами, та бібліотеку **AsyncUDP.h** для роботи із UDP.

Для того, аби сповістити користувача про можливу атаку на внутрішню мережу на дисплеї повинні з’являтися повідомлення, які відображають порт, до якого приходять пакети, а також IP-адресу джерела.

Обробка UDP пакетів буде виконуватись в функції **parseUDPPacket()**:

```
void parseUDPPacket(AsyncUDPPacket packet) {
  M5.Lcd.print("PORT:");
  M5.Lcd.print(packet.remotePort());
  M5.Lcd.print("|IP:");
  M5.Lcd.println(packet.remoteIP());
 }
```

У наведеній частині функції ми сповіщаємо користувача про те, що на наш пристрій прийшов новий пакет, виводимо на дисплей порт (packet.remotePort()), на який його було відправлено, а також IP-адресу джерела (packet.remoteIP()). Подібна обробка відбувається і з TCP пакетами в функції **parseTCPPacket()**:

```
void parseTCPPacket(WiFiClient client) {
  M5.Lcd.print("PORT:");
  M5.Lcd.print(client.remotePort());
  M5.Lcd.print("|IP:");
  M5.Lcd.println(client.remoteIP());
}
```

Функція **parseUDPPacket()** викликається із налаштувальної функції **setup()** в наступних рядках коду:

```
if (udp.listen(port)) {
  udp.onPacket(parseUDPPacket);
}
```

А виклик **parseTCPPacket()** реалізовано в зацикленій функції **loop()** та має наступний вигляд:

```
WiFiClient client = myServer.available();
if (client) {
  parseTCPPacket(client);
}
```

Перший рядок створює змінну client, яка чекає на підключення клієнту і в подальшому, коли підключення відбудеться, буде мати в собі потрібні нам дані, які в свою чергу будуть використані в функції **parseTCPPacket()**.

Окремою проблемою було те, що при надходженні великої кількості пакетів, екран швидко заповнювався інформацією і дані про пакети, які надходили далі не було видно. Складалося враження, що пристрій перестав працювати належним чином, тому було прийнято рішення очищувати дисплей при його заповенні. Щоправда, такий підхід унеможливлює перегляд частини інорфації, яку було стерто, безпосередньо на пристрої. Проте, по суті, наш дисплей відіграє суто сигналізуючу функцію, а подальший аналіз атак буде відбуватися за допомогою перегляду сформованого логу, тому це не грає важливої ролі. У майбутньому, при модифікації приладу, можливий інший шлях вирішення цієї проблеми, а саме, реалізація прокрутки (скролінгу) тексту донизу.

Очистка дисплею реалізована за допомогою спеціальної змінної (**threshold**), яка повинна рахувати кількість відображених рядків.

```
if (threshold == 14) {
  M5.Lcd.clearDisplay();
  M5.Lcd.setCursor(0, 0);
  threshold = 0;
}
```

4.2.3 Розробка частини програми протоколювання

Частина здійснення протоколювання є чи не найважливішою у всій роботі, тому що саме вона дозволяє отримати найбільшу користь від роботи пристрою, а саме аналіз атак, які здійснювалися на мережу. Так як дані повинні мати конкретні поля і їх значення, доцільно викоритстати таку структуру даних, як **реляційна база даних**.

Реляційні бази даних представляють собою набір даних із певним зв'язком. Дані організовні у вигляді таблиць, які складаються із рядків і стовпчиків. Для керування БД використовуються так звані системи керування базами даних, або **СУБД**. Сьогодні існує досить багато варіантів таких систем, найпопулярніші із них: Microsoft SQL Server, Oracle Database, MySQL and IBM DB2. Всі вони мають свої особливості та переваги перед іншими, але в цілому вони створенні для однієї цілі — керування базами даних.

У даному проєкті в нас немає ніяких специфічних задач, тому можна обрати будь-яку. Для себе я обрав **MySQL** через простоту роботи з нею, а також через те, що вже мав деякий досвід роботи з нею.

Тепер треба було вирішити яким чином ESP32 буде контактувати із БД та відправляти туди потрібні дані. Зробити це можна за допомогою поєднання спеціального вебдодатку **phpMyAdmin** та розгорнутого вебсерверу, наприклад **Apache**, куди ми будемо робити спеціальні запити.

phpMyAdmin створений спеціально для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних.

Apache ("Апачі", Apache HTTP Server) — це відкрите крос-платформне програмне забезпечення для розміщення та підтримки веб-сервера. Цей варіант є оптимальним для поставленої цілі, так як Апачі дозволяє відкривати динамічні PHP-сторінки, розподіляти навантаження, що надходить на сервер, а також запускати PHP-скрипти, що ми і збираємось робити.

Отже, мінімально необхідний набір софту для нас включає в себе: ОС, де буде здійснюватись все налаштування, найкраще підійде Linux, сервер, куди ESP32 буде здійснювати HTTP POST запити, передаючи інформацію про пакети, систему керування базами даних MySQL, а також встановлений PHP. Такий набір використовується доволі часто, існує навіть спеціальний акронім **LAMP** (Linux, Apache, MySQL, PHP).

У першу чергу, потрібно розгорнути сервер, зробити це можна скориставшись службою хостингу, якщо ми хочемо зробити можливим доступ із зовнішньої мережі, або ж встановити Apache на локальній машині і звузити коло доступа до пристроїв у внутрішній мережі. Перший варіант був би зручнішим, але це коштує грошей, тому для демонстрації я скористаюсь ручним варіантом налаштування.

На даному етапі також існує декілька варіантів:

- встановити Linux на свою локальну машину у якості другої ОС
- встановити Linux на віртуальну машину
- скористатися додатковим приладом, наприклад, Raspberry Pi, який працює на **Raspberry Pi OS** (заснована на Debian операційна система)
- скористатися Docker (інструментарій для управління ізольованими Linux-контейнерами)

Для себе я знаходжу останній варіант найбільш привабливим, так як знайшов готовий образ із необхідним мені набором LAMP і все, що мені треба зробити, це завантажити його та розгорнути в докері. Зробивши це, можемо перейти за адресою **http://localhost/** і впевнитись в тому, що сервер запрацював. Ми повинні побачити вікно із інформацією про версії програмного забезпечення, яке використовується в нашому докер контейнері (рис. 4.12).

Рисунок 4.12 — початкова сторінка, яка підтверджує, що Апачі працює

Тепер, коли сервер розгорнуто можна приступати до налаштування БД. За замовчуванням образ постачається з обліковим записом root MySQL без пароля. Цей обліковий запис доступний лише локально, тобто тільки всередині програми. Він недоступний з-поза образу докера або через phpMyAdmin.

Коли ви вперше запускаєте зображення, буде відображено повідомлення паролем вашого користувача (admin). Цього користувача можна використовувати як локально, так і ззовні, підключившись до порту MySQL (за замовчуванням 3306) і використовуючи такий інструмент, як MySQL Workbench або Sequel Pro, або через phpMyAdmin.

Отже, щоб налаштувати БД є два способи: через phpMyAdmin та через командний рядом. Перший варіант зручніший через наявність графічного інтерфейсу, тому краще використовувати його. Я створив нову БД та таблицю в ній (рис. 4.13).

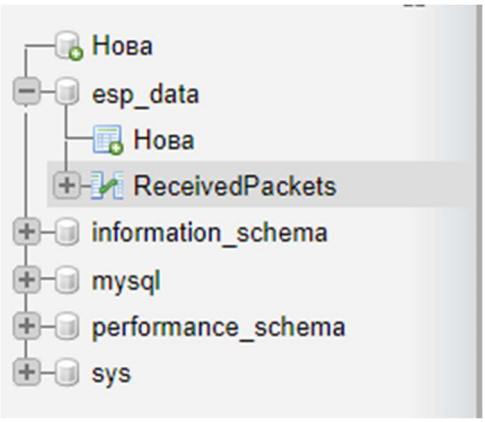

Рисунок 4.13 — створені БД “esp_data” та таблиця “ReceivedPackets”

Щоб створити таблицю із потрібними рядками я скориставсь наступним SQL запитом:

```
CREATE TABLE ReceivedPackets (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    protocol VARCHAR(5) NOT NULL,
    port int(11) NOT NULL,
    IP_address VARCHAR(15),
    received_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)
```

Наступним кроком буде створення PHP скріпта, який дозволить нам обробляти HTTP POST запити та поміщати дані із них в створену таблицю. Знайшовши потрібну директорію (/**app**), де розміщуються файли для відображення сторінок, створюємо новий файл під назвою ***post-esp-data.php*** із вмістом із додатку 2. Цей скріпт буде приймати POST запити, приймати з них значення протоколу, порту і IP адреси (значення ID та часу отримання пакету будуть генеруватися автоматично), підключатися до БД та створювати відповідний запит до неї для здійснення вставки:

```
1.$sql = "INSERT INTO ReceivedPackets (protocol, port, IP_address)
2.VALUES ('" . $protocol . "', '" . $port . "', '" . $IP_address . "')";
```

Крім цього, потрібно створити PHP файл, який буде коректно відображати дані із таблиці, для цього створимо ще один файл **esp-data.php** та наповнимо

його вмістом із додатку 3. Даний файл створить ендпоїнт, перейшовши на який користувач буде бачити вміст таблиці (рис. 4.15).

Рисунок 4.15 — відображення таблиці в браузері по ендпоїнту esp-data.php

Залишилось тільки змусити ESP32 робити POST запит на post-esp-data.php ендпоїнт у разі отримання пакету. Для цієї цілі було написано нову функцію sendDataToDatabase(). А також додано виклики цієї функції в parseTCPPacket() та parseUDPPacket().

```
void sendDataToDatabase(String protocol, uint16_t port, String ip_addr) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;
    HTTPClient http;
    // Your Domain name with URL path or IP address with path
    http.begin(client, serverName);

    // Specify content-type header
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    // Prepare your HTTP POST request data
    String httpRequestData = "api_key=" + apiKeyValue + "&protocol=" + protocol +
"&port=" + port + "&IP_address=" + ip_addr;

    int httpResponseCode = http.POST(httpRequestData);
  }
}
```

Тепер все готово і ми зможемо оперувати даними, які приходять із M5Stack до нашої бази даних за допомогою phpMyAdmin. Використовуючи структуровано мову запитів (SQL, structured query language) ми можемо

фільтрувати записи по конкретним портам, або по діапазону портів, по часу отримання, по протоколу, групувати ці дані як завгодно та інше.

Отже, поставлену задачу виконано, проте як і будь-який проєкт, дана робота ще не кінцевий результат і буде змінюватись в ході розробки, у майбутньому будуть з’являтися нові функції та можливості, такі як провокація зловмисників на підбор фіктивного паролю, провокація на передачу троянської програми на фіктивний диск, поліпшення візуальної складової у вигляді модернізації способу виведення тексту на дисплей, можливість пристрою розрізняти види атак (сканування мережі, підбор паролю SSH/FTP, спроби анонімної автентифікації та ін.) можлива реалізація функціоналу для виведення рекомендацій по захисту мережі в залежності від типу атаки та багато чого іншого.

5 ІСПИТ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

5.1 Програма іспиту

У програму іспиту входить тестування всіх частин проєкту:

- коректне відображення повідомлень на дисплеї M5Stack після отримання пакету;
- здатність приладу працювати із протоколами TCP та UDP;
- тестування серверу та його здатності обробляти HTTP POST запити;
- здатність M5Stack робити POST запити на вказану адресу;
- коректне відображення вмісту бази даних у браузері.

На першому кроці потрібно згенерувати декілька пакетів та відправити їх на M5Stack, при цьому, гарною ідеєю було б використати аналізатор трафіка та впевнитись, що пакети дійсно були відправлені, а також коротко переглянути їх вміст і порівняти його із відображеним на дисплеї.

Щоб впевнитись у здатності приладу працювати із обома найпоширенішими інтернет протоколам (TCP та UDP) достатньо буде, як і в першому етапі перевірки, згенерувати декілька пакетів TCP та UDP і подивитися на «реакцію» M5Stack.

Далі необхідно перевірити, чи відпрацьовує сервер свою задачу і чи справді він може обробляти запити та поміщати їх вміст у базу даних. Доволі зручним способом зробити це, використати спеціальну платформу для тестування серверів, де можна, в зручний шлях, згенерувати свій власний запит та побачити відповідь сервера.

Важливим етапом проєкту є здатність зберігати дані про можливі атаки на внутрішню мережу. Так як в роботі це реалізовано за допомогою створення M5Stack HTTP запиту, потрібно бути впевненим що прилад дійсно може звертатися до серверу і передавати йому дані.

Зручний перегляд збережених даних відіграє важливу роль для майбутнього користувача. Тому обрана реалізація, яка не вимагає він нього встановлення додаткового софту або будь-якого налаштування є великим плюсом. Важливо перевірити точність та коректність відображуваних даних.

5.2 Обладнання іспиту

У якості обладнання в першу чергу я маю M5Stack Core на базі ESP32, який оснащений дисплеєм та вайфай модулем для підключення до бездротової мережі, звичайний домашній роутер D-Link моделі DIR-615, до якого підключені всі прилади, а також мій персональний комп’ютер, який виступає одночасно в ролі серверу, де розгорнутий Apache, phpMyAdmin та mySQL, а також в ролі користувацького ПК.

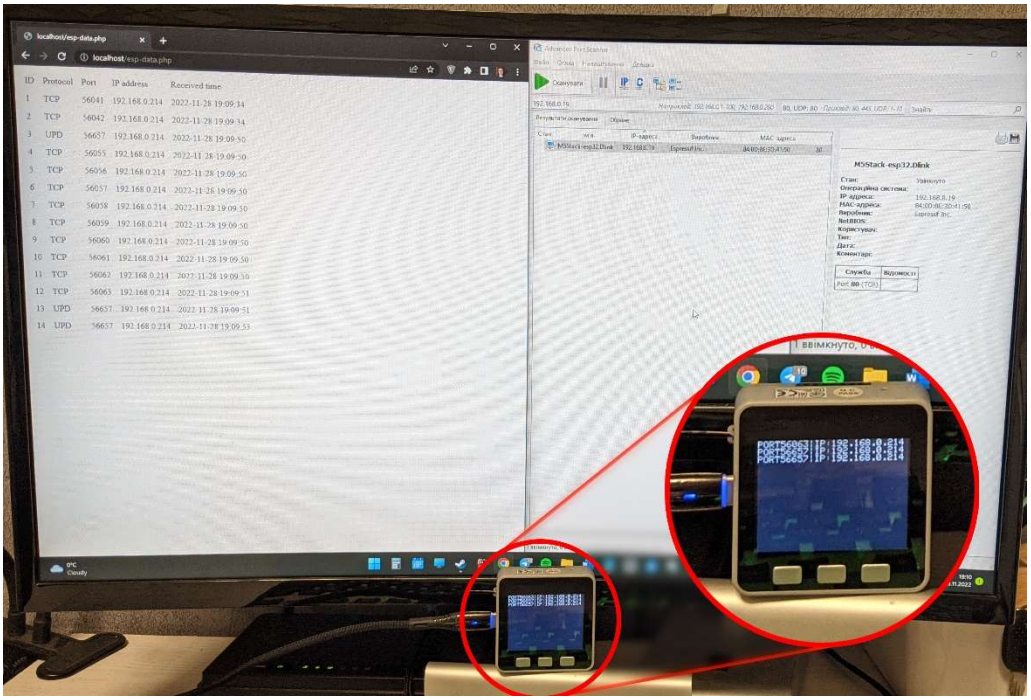

Рисунок 5.1 — макет для проведення іспитів

5.3 Результати іспиту

Для того, щоб перевірити відображення повідомлень на екрані M5Stack я скористаюсь утілітою nmap та згенерую декілька пакетів на локальну IP адресу нашого приладу (рис. 5.2):

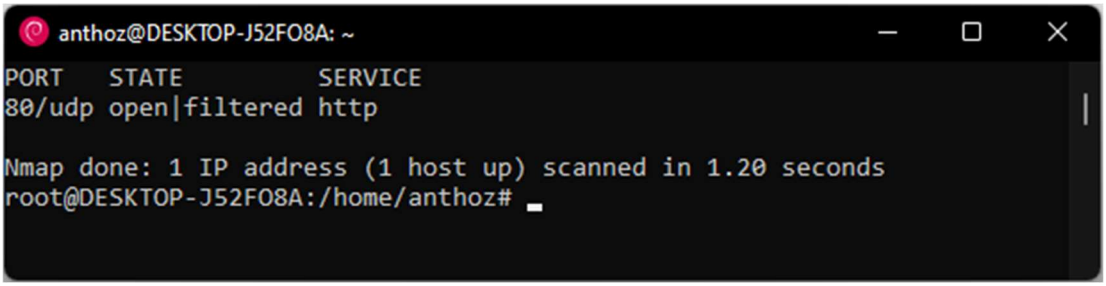

Рисунок 5.2 — генерація тестового пакету за допомогою Nmap

Також я скористаюсь програмою для аналізу трафіку Wireshark аби подивитись чи дійсно пакети були відправлені (рис. 5.3):

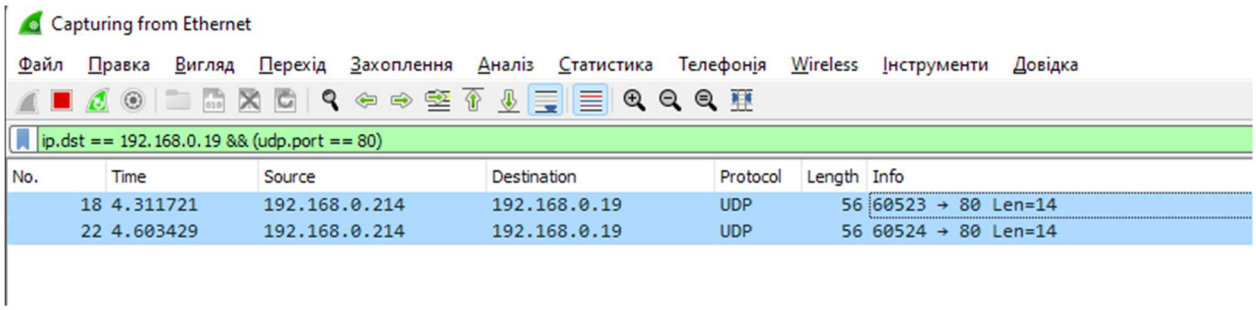

Рисунок 5.3 — інформація, відображена в програмі-аналізаторі

Бачимо, що було згенеровано два пакети з локальної IP адреси мого комп’ютера на локальну адресу приладу, порти з яких були передані повідомлення — 60523, 60524. Потрібно пам’ятати, що в реальних умовах буде відбуватися проброс портів, тобто повідомлення, яке, наприклад, приходить на 23 порт ком’ютера, після пробросу, в **port source (remote port)** буде також мати 23 порт, таким чином ми побачимо який порт сканувався (рис. 5.4).

Рисунок 5.4 — зміна порту джерела під час пробросу портів

На дисплеї приладу бачимо наступне:

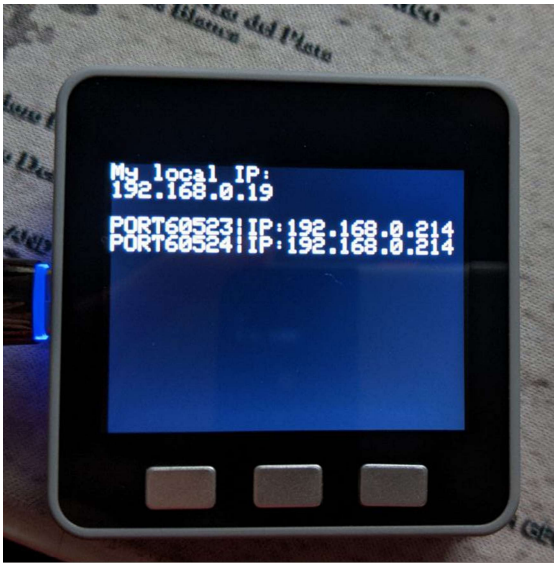

Рисунок 5.5 — відобрежння даних на дисплеї M5Stack

Бачимо локальну адресу приладу (192.168.0.19), на яку було відправлено пакети, а також два нових повідомлення, де вказано значення віддалених порту та IP адреси. Отже, робимо висновок, що сповіщення методом відобрежння інформації на дисплеї працює так, як було задумано.

Для того, аби впевнитись, що приладом приймаються та оброблюються пакети як UDP, так і TCP, я скористаюсь nmap із Windows PowerShall та згенерую декілька пакетів різного протоколу:

```
Nmap done: 1 IP address (1 host up) scanned in 0.56 seconds
PS C:\Users\Anthoz> nmap -sT -p 80 192.168.0.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-16 11:59 FLE Standard Time
Nmap scan report for M5Stack-esp32.Dlink (192.168.0.19)
Host is up (0.044s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 84:0D:8E:3D:41:50 (Espressif)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
PS C:\Users\Anthoz> nmap -sU -p 80 192.168.0.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-16 11:59 FLE Standard Time
Nmap scan report for M5Stack-esp32.Dlink (192.168.0.19)
Host is up (0.061s latency).

PORT   STATE         SERVICE
80/udp open|filtered http
MAC Address: 84:0D:8E:3D:41:50 (Espressif)
```

Рисунок 5.6 — створення TCP та UDP пакетів

Так як команда **nmap -sU 80 192.168.0.19** згенерує та відправить два UDP пакети, а **nmap -sT 80 192.168.0.19** один пакет TCP, в сумі на наш прилад повинно було прийти пакети. Можемо перевірити це, поглянувши в нашу базу даних, куди зберігаються всі записи (рис. 5.7):

```
SELECT * FROM `ReceivedPackets`;
```

Профілювання [Порядкове редагування] [Редагувати] [Тлумачити SQL] [Створити PHP код] [Оновити]

Показати все | Число рядків: 25 Фільтрувати рядки: Шукати в таблиці Сортувати за ключем: Жодного

+ Параметри

←T→			id	protocol	port	IP_address	received_time
Редагувати	Копіювати	Видалити	1	TCP	28272	192.168.0.214	2022-11-16 09:59:41
Редагувати	Копіювати	Видалити	2	UPD	39320	192.168.0.214	2022-11-16 09:59:45
Редагувати	Копіювати	Видалити	3	UPD	39322	192.168.0.214	2022-11-16 09:59:45

Рисунок 5.7 — вигляд вмісту бази даних

Всередині дійсно бачимо три пакети, 2 UDP та 1 TCP, IP адеса збігається із локальною адресою мого комп’ютера, звідки і було проведено тестування, отже все працює коректно.

Щоб протестувати сервер, я скористаюся спеціальним інструментом для тестування Postman, за допомогою якого можна створювати HTTP запити та відслідковувати відповідь сервера.

Щоб сформувати потрібний запит достатньо ввести адресу, на яку його буде відправлено, в нашому випадку це ***localhost:80/post-esp-data.php*** обрати тип **POST** та в тілі метода вписати вміст, відповідно, до заданого шаблону який має оброблятися PHP скріптом. У полі **api_key** повинен бути ключ, який співпадає із значенням відповідної змінної, яка задається в оброблювальному скріпті. У значення **protocol** впишемо TCP, **port** — 22, **IP_address** — 192.168.0.15**.**

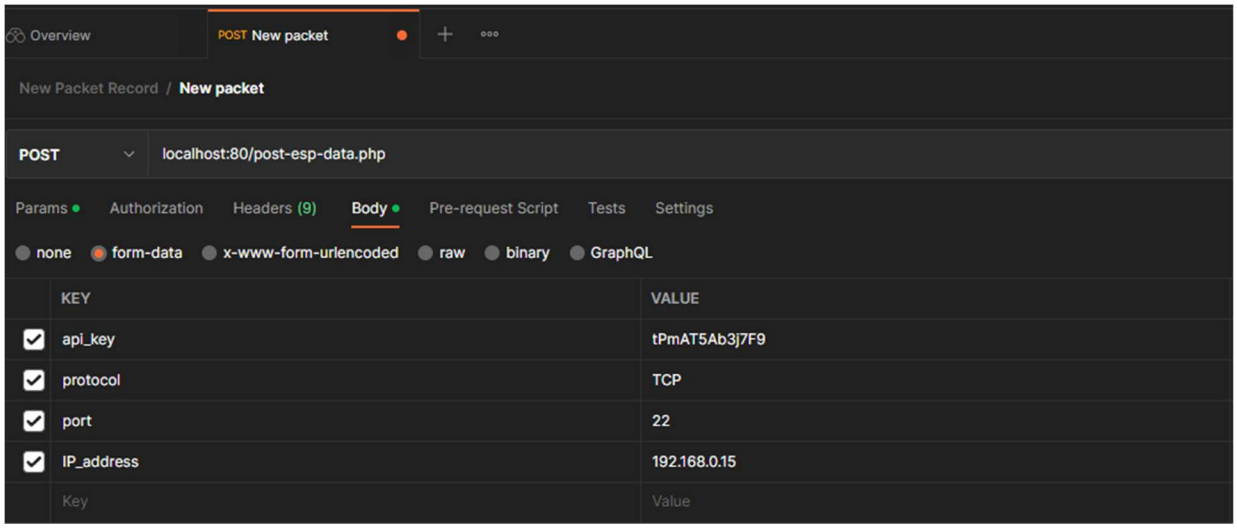

Рисунок 5.8 — створення запиту в Postman

Натиснувши кнопку Send, ми можемо побачити відповідь сервера, що прийшла після нашого запиту (рис. 5.9).

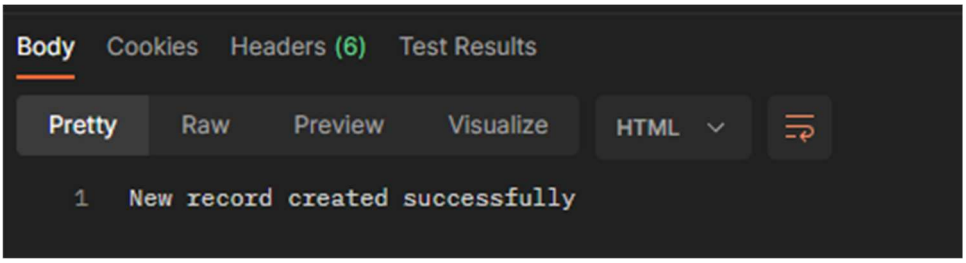

Рисунок 5.9 — повідомлення про успішну обробку запиту сервером

Також, в базі даних з’явився новий запис із даними, які були вказані в тестовому пакеті, що свідчить про те, що сервер повністю справляється із своєю задачею.

Наступним кроком перевірим здастність M5Stack відправляти HTTP запити до серверу. Для цього я напишу невелику програму, де буде міститися цикл із сотнею ітерацій, кожна з яких буде робити тестовий запит, очищу базу даних від усіх попередніх записів. Якщо все спрацює правильно, то в підсумку в БД буде міститися 100 рядків. Крім здатності M5Stack робити запити, заразом перевірим можливість серверу приймати велику кількість даних за короткий проміжок часу.

Після завантаження коду на M5Stack в базі даних побачимо рівно сто записів із вмістом, який було використано в тестових запитах.

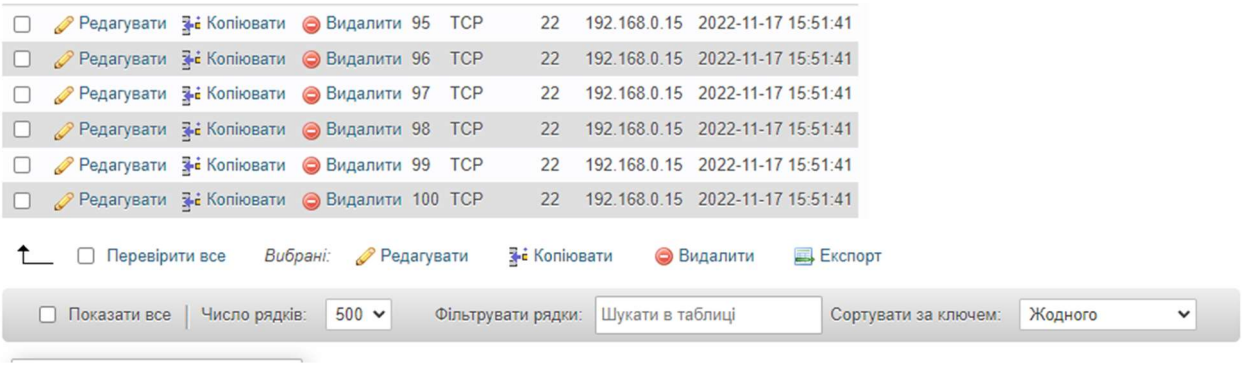

Рисунок 5.10 — вміст БД після тестування на обробку великої кількості запитів від M5Stack

За складеною програмою іспиту залишилось перевірити тільки коректнітсь відображення інформації із бази даних в браузері. Отже, за умови, що ми не очищували БД із поперннього тесту, наразі там повинно бути сто записів із однаковим вмістом. Перйдемо за адресою localhost/data-esp.php та поглянемо чи дійсно це так.

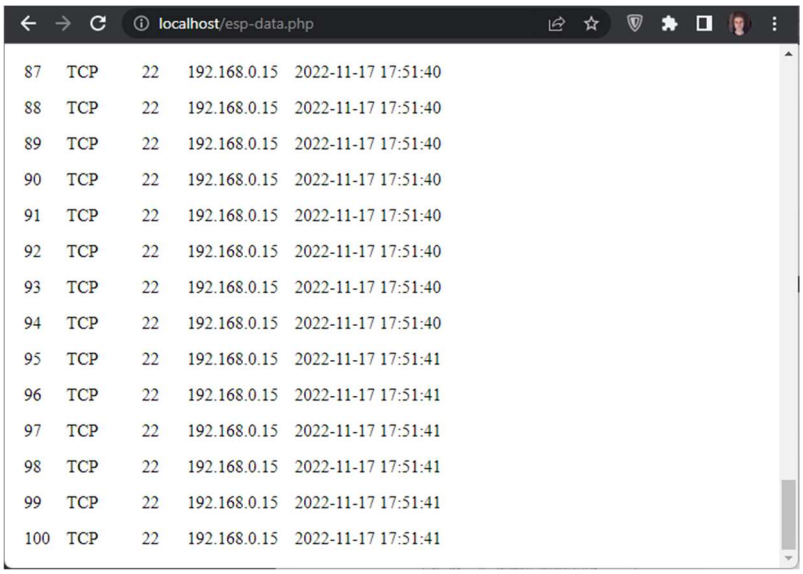

Рисунок 5.11 — відобреження вмісту БД в браузері

Як бачимо, в браузері дійсно відображено сотню записів із однаковим вмістом (за виключенням часу, за зрозумілих причин). Із цього можна зробити висновок, що відображення вмісту бази даних відповідає дійсності.

Отже, провівши тести по всім пунктам із програми іспиту, ми можемо побачити, що прилад цілком справляється із всіми поставленими задачами, а саме: з відображенням інформації на дисплеї приладу у разі прийнятого пакету, із здатністю M5Stack обробляти пакети обох основних протоколів (TCP та UDP), а також здатності приладу самому створювати HTTP запити із певним вмістом та відправляти їх на вказаний сервер. Також було протестовано сервер на здатність обробляти велику кількість запитів у короткий проміжок часу і коректність відображення збережених в БД даних у браузері.

6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

6.1 Маркетинговий аналіз стартап-проєкту

Ідея проєкту полягає в створенні програмно-апаратного комплексу протоколювання та сповіщення про атаку на внутрішню мережу. Комплекс представляє з себе невеликий прилад, розмірами 550 х 550 мм, який підключається до бездротової мережі Wi-Fi, його задачею є відслідковувати факт нападу на внутрішню мережу (на роутер).

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Створення програмно-апаратного комплексу протоколювання та сповіщення про атаку на внутрішню мережу	1. Кібербезпека організацій	Підвищення рівня безпеки корпоративної мережі
	2. Кібербезпека окремих осіб	Захист своєї домашньої мережі від спроб зловмисників її зламати

Під час таких атак на роутер приходять різного роду запити, за допомогою яких хакери здатні здійснювати кібератаки, при підключеному приладі, такі запити перехоплюються та оброблюються, при цьому відомості про них зберігються у базі диних, де їх можна буде проаналізувати у майбутньому, користувача в момент атаки прилад сповіщає про можливу небезпеку, даючи можливість вчасно відреагувати.

Аналізуючи ринкові можливості щодо реалізації проєкту, я дійшов висновку, що найкраще розробити його на основі вже готового продукту M5Stack, який представляє із себе невеликий комп'ютерний пристрій, котрий містить в собі необхідні функції для розробки прототипу, такі як бездротовий зв'язок, дисплей, а також кнопки керування.

Провівши дослідження, я виявив, що подібних приладів на ринку поки не існує. Конкуренцію можуть складати тільки спеціалізовані програми які

дозволяють захиститись від сканування портів, проте вони не надають ніякої додаткової інформації: звідки проводилося сканування, які порти сканувалися і ін.

Прилад розрахований на використання організаціями малого та середнього бізнесу, хоча я також розглядаю, сценарії використання даного приладу звичайними користувачами, які хотіли б мати більше інформації про те, що відбувається в їхній мережі.

Таблиця 6.2. Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
	Захист внутрішньої мережі від здійснення кібератак	Організації малого та середнього бізнесу, окремі особи	У випадку використання приладу організаціями, вони мають мати спеціаліста, який би правильно налаштував роутер для коректної роботи. При використанні приладу окремою особою, вона повинна мати деякі знання мережевих технлогій так як налаштування буде покладено на неї	Вимогами споживачів до продукції виступають надійність, точність та достовірність отримуваної від приладу інформації Від компанії-постачальника споживач повинен розраховувати на підтримку у разі непередбачуваних ситуацій або некоректної роботи пристрою

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities на

основі виділених ринкових загроз та можливостей, та сильних і слабких сторін.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.3. SWOT- аналіз стартап-проекту

Сильні сторони: відсутність прямої конкуренції	Слабкі сторони: відносно вузька направленість девайсу
Можливості: широка популярність через збільшення числа кібернетичних атак	Загрози: поява спеціалізованого ПЗ із схожим функціоналом

6.2 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування *маркетингової концепції товару*, який отримає споживач.

Таблиця 6.4. Визначення ключових переваг концепції потенційного товару

Потреба	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
Підтримка безпеки інформаційних даних та захист внутрішньої мережі	Аналіз даних, які були зібрані під час спроб атак. Відсутність необхідності встановлювати додатковий софт	Прилад виступає автономною одиницею, тобто не потребує включення комп'ютера, що є перевагою перед конкурентними програмами із схожими функціями. Також прилад надає можливість самостійно аналізувати дані про надходження небажаних пакетів на адресу роутера

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу

проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів.

Таблиця 6.5. Визначення меж встановлення ціни

Рівень цін на товари-замінники	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
Щомісячний платіж ~15 $ (у випадку заміни спеціалізованими програмами)	-	500-1000$	65-80$

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення. На мою думку краще проводити збут власними силами без залучення сторонніх посередників, у цілях зменшення витрат та запобігання недотримання перевізниками правил транспортування подібних приладів.

6.3 Висновки до стартап-проєкту

Отже, у ході розробки стартап-проєкту я дійшов висновку, що ринкова комерціалізація проєкту дійсно можлива, через такі фактори як наявність попиту, позитивну динаміку росту ринку та хороший рівень рентабельності роботи. Також я вбачаю перспективи впровадження з огляду на потенційні групи клієнтів, зокрема через цифровізацію та збільшення кількості віддаленої роботи. Я вважаю подальшу імплментацію проєкту цілком доцільною.

ВИСНОВКИ

Задачею, яка ставилася перед собою було створення програмно-апаратного комплексу сповіщення про атаки на внутрішню мережу. Пристрій такого роду допоможе не тільки попередити атаку на мережу, але і допомогти зменшити кількість випадків такого типу кібератак у майбутньому.

Враховуючи те, що про факт більшісті кібератак користувачі внутрішніх мереж дізнаються тільки у разі вже нанесеної шкоди та збитків, актуальність розробки системи про попередження про факт спроби кібератаки сумеівів не визиває.

У ході даної роботи було проведено окреме дослідження різноманітних способів кібератак, їх види та способи реалізацій, а також сучасну тенденцію росту таких атак. Було проведено дослідження способів проникнення у внутрішню мережу. Посилання на численні публікації, що були використані у цьому обзорі предоставлені у списку літератури.

Було встановлено, що для більшості типів кібератак з метою проникнення у внутрішню мережу спільним є сканування портів пристрою доступу до мережі. Представлений список найбільш "небезпечних" портів та їх призначень, з акцентуванням уваги на ті, які потребують найбільше уваги із сторони користувача, також розглянуто сценарії протидії злому.

Проведені дослідження дозволили сформувати вимоги для програмно-апаратного комплексу попередження про кібератаки та протоколювання дій зловмисників. На основі цих вимог сформоване ТЗ. На основі якого був створений діючий прототип цього комплексу.

Прототип створений на основі мікроконтролеру M5Stack, який здатен оброблювати перехоплені TCP та UDP пакети, сповіщати користувача проможливу спробу атаки, відображуючи повідомлення на дисплеї. А також створювати записи в, спеціально створеній, базі даних. Безперечною перевагою такого інженерного рішення є те, що доступ до вмісту бази данних

про кібератаки можна отримати без встановлення будь-яких додаткових програм на вашу машину, потрібна лише наявність браузера. Також, користувач має можливіть переглядати та аналізувати дані всередині бази даних за допомогою структурованої мови запитів (SQL).

Проведений успішний іспит створеного прототипу. Предоставлені результати іспиту, включаючи фото прототипу, та скріни попереджень про атаки.

Експериментально довелена працездатність створеного програмно-апаратного комплексу та корректність вибраних інженерних рішень. Розроблені програми наведені у додатку до роботи.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Predicting What 2022 Holds For Cybersecurity [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.forbes.com/sites/emilsayegh/2022/01/06/predicting-what-2022-holds-for-cybersecurity/?sh=4c66a852b728.
2. Cybercrime To Cost The World $10.5 Trillion Annually By 2025 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/.
3. Data Breach Investigations Report [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.verizon.com/business/resources/reports/2020-data-breach-investigations-report.pdf.
4. Average duration of downtime after a ransomware attack from 1st quarter 2020 to 4th quarter 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.statista.com/statistics/1275029/length-of-downtime-after-ransomware-attack/.
5. The damaging impact of cybercrime on executive leadership [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://info.constellaintelligence.com/hubfs/Getting-Personal-Exec-Protection-Report-WP-2021.pdf
6. Healthcare data breaches hit all-time high in 2021, impacting 45M people [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.fiercehealthcare.com/health-tech/healthcare-data-breaches-hit-all-time-high-2021-impacting-45m-people.
7. Report: Ransomware attacks cost local and state governments over $18 billion in 2020 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: 10.

https://www.americancityandcounty.com/2021/03/22/report-ransomware-attacks-cost-local-and-state-governments-over-18-billion-in-2020/.

8. Report: Ransomware attacks cost local and state governments over $18 billion in 2020 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.americancityandcounty.com/2021/03/22/report-ransomware-attacks-cost-local-and-state-governments-over-18-billion-in-2020/.
9. Municipal Cyberattacks: A New Threat Or Persistent Risk? [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.forbes.com/sites/forbestechcouncil/2021/06/22/municipal-cyberattacks-a-new-threat-or-persistent-risk/?sh=295349e13ffb.
10. Cybersecurity in Non-Profit and Non-Governmental Organizations [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://icitech.org/wp-content/uploads/2017/02/ICIT-Brif-Cybersecurity-and-NGOs.pdf.
11. Hackers Sell Backdoors Into A $2 Billion Nonprofit, A Californian Hospital, And Michigan Government [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.forbes.com/sites/thomasbrewster/2022/02/23/hackers-sell-access-to-a-2-billion-nonprofit-a-californian-hospital-and-michigan-government/?sh=2e1caacc5758.
12. Cyber-attack on ICRC: What we know [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.icrc.org/en/document/cyber-attack-icrc-what-we-know.
13. X-Force Threat Intelligence Index 2022 [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.ibm.com/downloads/cas/ADLMYLAZ.
14. Worry-Free Business Security Services [Електронний ресурс] – Режим доступу до ресурсу: https://docs.trendmicro.com/all/smb/wfbs-

services/Server/Dell/v3.7/ru/docs/WebHelp/_WFBS-SVC/C07-ConfiguringGroupSecuritySettings/IntrusionDetectionSystem.htm.

15. FBI Alerts Medical and Dental Facilities on the Anonymous FTP Security Vulnerability [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://www.globalscape.com/blog/fbi-alerts-medical-and-dental-facilities-anonymous-ftp-security-vulnerability.
16. Список незахищеної захищеної інформації про здоров'я [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf.
17. Github[samyk] - NAT Slipstreaming [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://github.com/samyk/slipstream
18. ALERT LOGIC CRITICAL WATCH ANALYSIS BY THE NUMBERS [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.alertlogic.com/assets/critcal-watch-report/CWR19_ExecReport.pdf
19. The Future of Hybrid Work: 5 Key Questions Answered With Data [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: 19. https://www.gallup.com/workplace/390632/future-hybrid-work-key-questions-answered-data.aspx.

ДОДАТОК 1

Код, призначений для демонстрації вразливості до перебору паролів Telnet (23 порт).

Мова програмування: Python.

```
import telnetlib

TIMEOUT = 4
counter = 0
ip = input("Enter ip-address: ")
port = int(input("Enter Telnet Port: "))
login = input("Enter Login: ")
user = login + '\n'
flag = True

while flag:

    try:
        tn = telnetlib.Telnet(ip, port)
        tn.read_until(b'Dlink-Router login:')
        tn.write(user.encode('ascii'))
        tn.read_until(b'Password:')
        with open('password.txt', 'r') as password:
            passwd = password.read().splitlines()

        passmain = passwd[counter] + '\n'
        tn.write(passmain.encode('ascii'))

        if tn.expect([b'Login incorrect'], TIMEOUT)[0] == -1:
            print("Password found!")
            print(login, "and", passmain)
            flag = False
            tn.close()

        else:
            print(passmain)

        counter += 1

    except:
        print("Telnet port is closed!")
        flag = False
```

ДОДАТОК 2

Вміст файлу post-esp-data.php: код PHP-скріпта, призначеного для обробки POST запитів.

Мова програмування: PHP.

```
1.<?php
2.
3.$servername = "localhost";
4.
5.// REPLACE with your Database name
6.$dbname = "esp_data";
7.// REPLACE with Database user
8.$username = "admin";
9.// REPLACE with Database user password
10.$password = "emjZqQSNfN9d";
11.
12.// Keep this API Key value to be compatible with the ESP32 code provided
   in the project page.
13.// If you change this value, the ESP32 sketch needs to match
14.$api_key_value = "tPmAT5Ab3j7F9";
15.
16.$api_key = $protocol = $port = $IP_address = "";
17.
18.if ($_SERVER["REQUEST_METHOD"] == "POST") {
19.    $api_key = test_input($_POST["api_key"]);
20.    if($api_key == $api_key_value) {
21.        $protocol = test_input($_POST["protocol"]);
22.        $port = test_input($_POST["port"]);
23.        $IP_address = test_input($_POST["IP_address"]);
24.
25.        // Create connection
26.        $conn = new mysqli($servername, $username, $password, $dbname);
27.        // Check connection
28.        if ($conn->connect_error) {
29.            die("Connection failed: " . $conn->connect_error);
30.        }
31.
32.        $sql = "INSERT INTO ReceivedPackets (protocol, port, IP_address)
33.        VALUES ('" . $protocol . "', '" . $port . "', '" . $IP_address
   . "')";
34.
35.        if ($conn->query($sql) === TRUE) {
36.            echo "New record created successfully";
37.        }
38.        else {
39.            echo "Error: " . $sql . "<br>" . $conn->error;
40.        }
41.
42.        $conn->close();
43.    }
```

```
44.     else {
45.         echo "Wrong API Key provided.";
46.     }
47.
48.}
49.else {
50.     echo "No data posted with HTTP POST.";
51.}
52.
53.function test_input($data) {
54.     $data = trim($data);
55.     $data = stripslashes($data);
56.     $data = htmlspecialchars($data);
57.     return $data;
58.}
```

ДОДАТОК 3

Вміст файлу esp-data.php: код PHP-скріпта, призначеного для відображення вмісту бази даних у браузері.

Мова програмування: PHP.

```
1.<!DOCTYPE html>
2.<html><body>
3.<?php
4.
5.$servername = "localhost";
6.
7.// REPLACE with your Database name
8.$dbname = "esp_data";
9.// REPLACE with Database user
10.$username = "admin";
11.// REPLACE with Database user password
12.$password = "emjZqQSNfN9d";
13.
14.// Create connection
15.$conn = new mysqli($servername, $username, $password, $dbname);
16.// Check connection
17.if ($conn->connect_error) {
18.    die("Connection failed: " . $conn->connect_error);
19.}
20.
21.$sql = "SELECT id, protocol, port, IP_address, received_time FROM
   ReceivedPackets ORDER BY id";
22.
23.echo '<table cellspacing="5" cellpadding="5">
24.      <tr>
25.        <td>ID</td>
26.        <td>Protocol</td>
27.        <td>Port</td>
28.        <td>IP address</td>
29.        <td>Received time</td>
30.      </tr>';
31.
32.if ($result = $conn->query($sql)) {
33.    while ($row = $result->fetch_assoc()) {
34.        $row_id = $row["id"];
35.        $row_protocol = $row["protocol"];
36.        $row_port = $row["port"];
37.        $row_ip = $row["IP_address"];
38.        $row_time = $row["received_time"];
39.
40.        $row_time = date("Y-m-d H:i:s", strtotime("$row_time + 2 hours"));
41.
42.        echo '<tr>
43.                <td>' . $row_id . '</td>
44.                <td>' . $row_protocol . '</td>
```

```
45.                 <td>' . $row_port . '</td>
46.                 <td>' . $row_ip . '</td>
47.                 <td>' . $row_time . '</td>
48.               </tr>';
49.     }
50.     $result->free();
51.}
52.
53.$conn->close();
54.?>
55.</table>
56.</body>
57.</html>
```

ДОДАТОК 4

Код M5Stack, який дозволяє обробляти пакети, які приходять на нього через TCP або UDP протокол. Змінні port, ssid, password, serverName, apiKeyValue потрібно замінити на свої значення.

```
#include "M5Stack.h"

#include "WiFi.h"

#include "AsyncUDP.h"

#include <HTTPClient.h>

const uint16_t port = 80; // the port that will receive UDP packets after port
forwarding
AsyncUDP udp;
WiFiServer myServer(0);

void setup() {
  M5.Power.begin();
  M5.begin();

  M5.Lcd.setTextSize(2);

  initWiFi();

  myServer.begin(80); // set the port value for TCP packets

  if (udp.listen(port)) { // listen port for new connections (UDP)
    udp.onPacket(parseUDPPacket);
  }
}

// AP credentials
const char* ssid = "ENTER_YOUR_AP_NAME";
const char* password = "ENTER_YOUR_PASSWORD";

void initWiFi() { // method for WiFi connection
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  M5.Lcd.print("Connecting to WiFi..");

  while (WiFi.status() != WL_CONNECTED) {
    M5.Lcd.print('.');
    delay(1000);
```

```
  }

  M5.Lcd.clearDisplay();
  M5.Lcd.setCursor(0, 0);
  M5.Lcd.println("My local IP:");

  M5.Lcd.println(WiFi.localIP());
  M5.Lcd.println();
}

int threshold = 3;

void parseUDPPacket(AsyncUDPPacket packet) {
  if (threshold == 14) {
    M5.Lcd.clearDisplay();
    M5.Lcd.setCursor(0, 0);
    threshold = 0;
  }

  M5.Lcd.print("PORT");
  M5.Lcd.print(packet.remotePort());
  M5.Lcd.print("|IP:");
  M5.Lcd.println(packet.remoteIP());

  sendDataToDatabase("UPD", packet.remotePort(), packet.remoteIP().toString());

  Serial.println("\nUPD packet received");

  Serial.print("Local port: ");
  Serial.println(packet.remotePort());

  Serial.print("Remote port: ");
  Serial.println(packet.remotePort());

  Serial.print("Local IP: ");
  Serial.println(packet.localIP());

  Serial.print("Remote IP: ");
  Serial.println(packet.remoteIP().toString());
  threshold++;
}

void parseTCPPacket(WiFiClient client) {
  if (threshold == 14) {
    M5.Lcd.clearDisplay();
    M5.Lcd.setCursor(0, 0);
    threshold = 0;
  }

  M5.Lcd.print("PORT");
```

```
  M5.Lcd.print(client.remotePort());
  M5.Lcd.print("|IP:");
  M5.Lcd.println(client.remoteIP().toString());

  sendDataToDatabase("TCP", client.remotePort(), client.remoteIP().toString());

  Serial.println("\nTCP packet received");

  Serial.print("Local port: ");
  Serial.println(client.localPort());

  Serial.print("Remote IP: ");
  Serial.println(client.remoteIP());
  threshold++;
}

const char* serverName = "http://192.168.0.214/post-esp-data.php";  // server
endpoint processing HTTP POST requests
String apiKeyValue = "tPmAT5Ab3j7F9";                               // change
only when paired with post-esp-data.php!

void sendDataToDatabase(String protocol, uint16_t port, String ip_addr) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;
    HTTPClient http;
    // your Domain name with URL path or IP address with path
    http.begin(client, serverName);

    // specify content-type header
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    // Prepare your HTTP POST request data
    String httpRequestData = "api_key=" + apiKeyValue + "&protocol=" + protocol +
"&port=" + port + "&IP_address=" + ip_addr;

    int httpResponseCode = http.POST(httpRequestData);
  }
}

void loop() {
  WiFiClient client = myServer.available(); // listen for new connections (TCP)
  if (client && String(client.remoteIP()) != "127.0.0.1") {
    parseTCPPacket(client);
  }
}
```