МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Графічний інтерфейс для налаштування параметрів безпеки та аутентифікації користувача на маршрутизаторах Cisco»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Великодний Д.В.**

Студент групи ІК.м-91 — **Машутін А.Р.**

СУМИ 2020

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«Інформаційно-комунікаційні технології»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 2020 р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Машутіну Антону Русалновичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Графічний інтерфейс для налаштування параметрів безпеки та аутентифікації користувача на маршрутизаторах Cisco*
затверджую наказом по інституту від "____" _____ 2020 р. №_____

2. Термін здачі студентом закінченого проекту (роботи) _____
3. Вхідні данні до проекту (роботи) _____
_____
_____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз інформації. Постановка задачі дослідження. 2) Методологія захисту та безпеки маршрутизаторів Cisco. 3) Вивчення симулятору Cisco Packet Tracer. 4) Моделювання мережі. 5) Створення графічного інтерфейсу для налаштування маршрутизатора.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
_____
_____
_____
_____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	*Аналіз інформації. Постановка задачі дослідження.*		
2.	*Методологія захисту та безпеки маршрутизаторів Cisco.*		
3.	*Вивчення можливостей симулятору Cisco Packet Tracer.*		
4.	*Моделювання мережі.*		
5.	*Створення графічного інтерфейсу для налаштування маршрутизатора.*		
6.	*Оформлення пояснювальної записки.*		

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис)

РЕФЕРАТ

Записка: 60 стор., 18 рис., 3 додатки, 17 джерел.

Об’єкт дослідження — графічний інтерфейс для налаштування параметрів безпеки та аутентифікації користувача на маршрутизаторах Cisco.

Мета роботи — розробка графічного інтерфейсу для налаштування параметрів безпеки та аутентифікації користувача на маршрутизаторах Cisco.

Методи дослідження — в процесі роботи використано бібліотека jQuery, мови HTML, CSS, JavaScript, симулятор Cisco Packet Tracer.

Результати — проведено аналіз інформації, обрано методи розробки та дослідження, створені дизайн, програмна реалізація та виконано тестування.

ГРАФІЧНИЙ ІНТЕРФЕЙС, ВЕБ-САЙТ, ПАРАМЕТРИ БЕЗПЕКИ, МАРШРУТИЗАТОР, СИМУЛЯТОР МОДЕЛЮВАННЯ МЕРЕЖІ, ДИЗАЙН, КОМАНДНИЙ РЯДОК, ПАРОЛЬ, ТЕГ

Зміст

ВСТУП..... 6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД..... 7
1.1 Визначення та значення маршрутизатора в мережі..... 7
1.2 Необхідність налаштування безпеки..... 8
1.3 Види хакерських атак..... 9
1.4 Інтерфейс командного рядка..... 11
1.5 Способи захисту маршрутизатора..... 12
1.6 Постановка задачі..... 24
2 МОДЕЛЮВАННЯ МЕРЕЖІ ТА РОЗРОБКА ВЕБ-СТОРІНКИ ІНТЕРФЕЙСУ..... 25
2.1 Конфігурація мережі з використанням симулятору Cisco Packet Tracer..... 25
2.2 Розробка веб-сторінки інтерфейсу з використанням мови JavaScript..... 29
3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕРФЕЙСУ ДЛЯ НАЛАШТУВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ МАРШРУТИЗАТОРА..... 38
3.1 Веб-інтерфейс для налаштування безпеки та аутентифікації маршрутизаторів..... 38
3.2 Тестування графічного інтерфейсу в симуляторі Cisco Packet Tracer..... 42
ВИСНОВКИ..... 46
СПИСОК ЛІТЕРАТУРИ..... 47
ДОДАТОК А..... 49
ДОДАТОК Б..... 54
ДОДАТОК В..... 59

ВСТУП

У сфері телекомунікацій необхідними є такі речі, як безпека та аутентифікація. Адже наразі існує велика кількість різноманітних компаній та підприємств, які мають досить складні інформаційні мережі. Та з їх розвитком і ростом посилюються вимоги до надійності систем. Прикладами таких компаній можуть бути навчальні заклади, лікарні, промислові підприємства, магазини, комунальні заклади. Мережі, що використовуються в них називаються корпоративними. Вони повинні підтримувати обмін інформацією безпосередньо в самій мережі, а також підтримувати різні типи трафіку(відео, аудіо, електрона пошта, телефонія, файли з даними).

Для того, щоб мережа могла функціонувати стабільно необхідно досягти високого рівня надійності системи. Для цього компанії намагаються використовувати дороге та якісне обладнання, яке зможе виконати всі функції для покращення безпеки. Від того, яке обладнання використовується в мережі, залежить також і можливість передачі великої кількості інформації одночасно, відновлюваність системи та робота при виникненні аварійних ситуаціях. Проте зрозуміло, що лише купівля якісного обладнання не може гарантувати її повний захист. Бо найбільш важливим є правильне проектування мережі та її налаштування. Для того, щоб покращити захищеність системи необхідно розібратися в тому, які саме команди допоможуть це зробити. Адміністратор мережі повинен вміти налаштовувати обладнання та розуміти методи, які використовуються для цього. Також актуальність теми роботи підкріплюється тим, що попит на створення власних мереж зростає, компанії розширюють свої функціональні можливості, а значить виникає необхідність у налагодженні стабільної роботи всієї системи.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Визначення та значення маршрутизатора в мережі

Маршрутизатор — це вид мережевого обладнання, який приймає, фільтрує та відправляє пакети даних мережею. Він підключений як мінімум до двох, зазвичай до локальних або глобальних мереж чи до локальної та мережі свого провайдера. Залежно від пропускної здатності маршрутизатора він може передавати певний обсяг інформації. Він з'єднує пристрої в мережі один з одним і дозволяє іншим приладам передавати дані.

Існують різні типи маршрутизаторів. Вони бувають, як дротові, так і бездротові. Обидва передають пакети даних на комп'ютери та від них, але бездротові не можуть під'єднатися безпосередньо до комп'ютера через кабель. Замість цього вони використовують радіосигнал.

Маршрутизатори – це найбільш інтелектуальні пристрої в будь-якій мережі. Вони виконують дві важливі функції: вибір шляху та комутацію(процес з’єднання). Таблиці маршрутизації забезпечують виконання функції вибору оптимального шляху передавання пакетів. Функція комутації дозволяє з’єднати порт, з якого надійшов пакет, з портом, в який його потрібно відправити. Маршрутизатор запобігає виникненню зайвого трафіку в мережевих сегментах шляхом перевірки логічної адреси отримувача [1].

Однією з найбільших компаній, яка займається розробкою мережевого обладнання є Cisco Systems. Компанія надає своїм покупцям дуже широкий спектр різноманітного сучасного обладнання серед яких: маршрутизатори, комутатори, системи відеоспостереження, бездротові пристрої та інше. Можливості компанії безперервно розширюються. Наразі компанія охоплює велику кількість різних IT-галузей. Cisco Systems розробляє та продає системи безпеки, хмарні системи, програмне забезпечення, телефонію на базі Інтернету, постійно розвиває таку галузь, як інтернет речей та пропонує велику кількість освітніх програм з можливістю отримати високооплачувану роботу у сфері

телекомунікацій. Компанія створює велику кількість різних маршрутизаторів, кожен з яких направлений на виконання своєї функції. Наприклад, маршрутизатори такої серії, як Cisco 2800 підтримують одні з найбільш ефективних рішень у сфері IP-комунікацій. Починаючи від простої телефонії та закінчуючи такими функціями, як автоматична операторська служба, обробка мультимедійних викликів, система передачі повідомлень, які надають користувачам дуже широкі можливості для адаптації рішень під свої конкретні потреби. Маршрутизатор Cisco 2811 є частим вибором для використання в корпоративних мережах. Його фото зображено на Рис. 1.1 [2, 3].

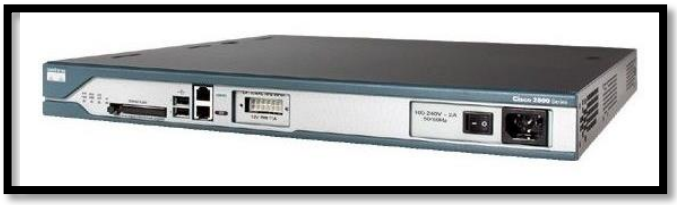

Рис. 1.1 – Cisco 2811

1.2 Необхідність налаштування безпеки

Після появи мережі Інтернет наше життя повністю змінилося. Те, як ми зараз живимо, працюємо та вчимося сильно відрізняється від того, як це робили лише 20 років тому. Проте разом зі значним зростанням популярності технології з’являється загроза розголошення особистих даних, надзвичайно важливої корпоративної інформації, таємниць держав та ін. Кожного дня зловмисники ставлять під загрозу ці ресурси. Вони намагаються отримати доступ використовуючи спеціально створені ними програми та додатки, які щоразу стають ще більш досконалими та простими в освоєнні. Цьому сильно сприяють декілька чинників.

Перший, це те, що надзвичайно популярним став Інтернет. Сьогодні до нього підключені мільярди пристроїв з усіх країн світу. А також ще мільйони приладів будуть підключені до мережі в майбутньому. Через це ймовірність того,

що хакери отримають можливість доступу до незахищених пристроїв постійно зростає. Крім того, надто широке поширення Інтернету дозволяє злодіям обмінюватися інформацією між собою без будь-яких наслідків.

Другий, це значне поширення нескладних у використанні програм, середовищ розробки та операційних систем. Саме це дозволяє значно знизити рівень необхідних знань для хакерів. Якщо порівнювати, то раніше така людина необхідна була володіти серйозними навичками в написанні коду для створення програм, які допомагають отримувати інформацію інших осіб. А тепер зловмисник має можливість надто легко здобути будь-які знання маючи лише доступ до браузера та адресу сайту з такими даними.

І через те, що маршрутизатор є надзвичайно важливим елементом будь-якої мережі, то отримавши можливість керувати ним можна спричинити великі грошові та інформаційні втрати.

1.3 Види хакерських атак

Коротко приведемо приклад часто використовуваних атак для того, щоб розуміти, що може статися в ситуаціях, коли вони проводяться.

Sniffer пакетів – це спеціальна програма, яка шляхом використання певних методів застосовує мережеву карту з метою перехоплення мережевих пакетів. Можливість отримання паролів та логінів користувача створює критичну небезпеку, бо часто застосовуються одні й ті ж самі паролі для різних ресурсів. При використанні такого способу заволодіння інформацією присутня можливість дізнатися конфіденційні дані людини чи організації [4].

Підміна адреси(IP spoofing) застосовується в ситуаціях, коли хакер, який знаходиться всередині мережі або зовні та видає себе за законного користувача. Зазвичай це робиться 2 способами. Перший, коли хакер має можливість використати IP-адресу, яка знаходиться в межах діапазону системи, або має таку зовнішню адресу, якій надається доступ до певних мережевих ресурсів. Атаки з використанням підміни адреси досить часто є початковою точкою для інших типів атак.

Відмова в обслуговуванні є однією з найбільш відомих методів хакерських нападів. Також можемо зазначити, що проти таких атак найважче створити повний стовідсотковий захист. А простота реалізації та надзвичайна шкода, яка завдається при її виконанні, залучають до вивчення DoS(Denial of Service) адміністраторами, що відповідають за мережеву безпеку мереж для розуміння способів боротьби з такими загрозами [5].

Простий перебір – один зі способів отримання інформації шляхом використання спеціальної програми, яка намагається отримати повний доступ до ресурсу загальнодоступного користування просто перебираючи паролі та логіни з метою пошуку збігів. А якщо в результаті таких дій хакер отримує доступ до потрібних ресурсів, то він дістає необхідні дані для входу в систему під виглядом звичайного користувача, конфіденційність якого була розкрита. Якщо такий користувач має привілеї адміністратора для доступу, то хакер може створити для себе можливість для майбутнього проникнення в систему, яка буде працювати, якщо власник навіть змінить свій логін та пароль.

Мережева розвідка — це накопичення інформації про систему з використанням загальнодоступних даних та додатків. При підготовці атаки проти будь-якої мережі злодій часто намагається дізнатися про неї якомога більше нового. Мережева розвідка проводиться у формі ping-запитів, DNS-запитів та сканування портів.

1.4 Інтерфейс командного рядка

Для того, щоб забезпечити надійний захист маршрутизатора та зменшити ймовірність вторгнення в мережу й отримання інформації необхідно використовувати різні методи для безпеки пристроїв. Для введення даних, необхідних для налаштування роутера, повинен бути використаний інтерфейс командного рядка(Command line interface).

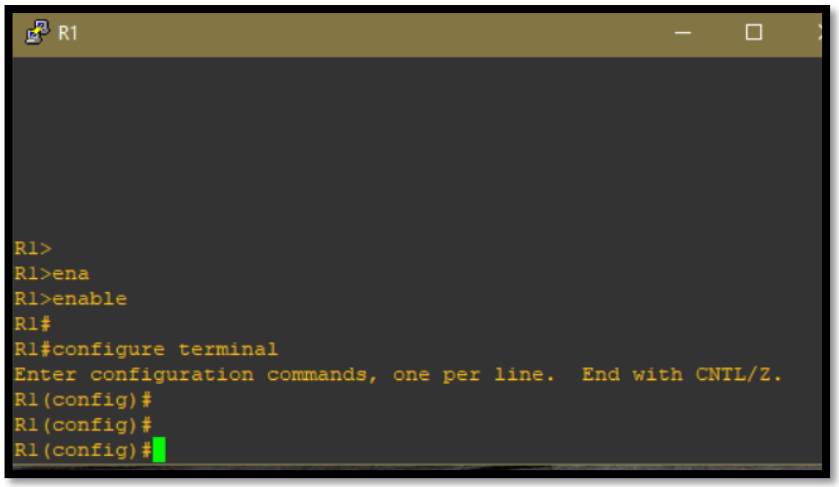

Рис. 1.2 – Вигляд CLI маршрутизаторів Cisco

Інтерфейс командного рядка (CLI) Cisco IOS — це основний інтерфейс, який використовується для налаштування, обслуговування та моніторингу обладнання Cisco. Він дозволяє одночасно виконувати команди за допомогою консолі маршрутизатора, терміналу або з використанням віддаленого доступу.

В Cisco IOS існує 2 рівні доступу — user exec mode, тобто доступ призначений для користувача та privileged exec mode, що означає режим привілейованого доступу [6].

Коли маршрутизатор вмикається, то режим доступу користувача встановлюється за замовчуванням. У цьому режимі в командному рядку показується така підказка:

Router>

У цьому режимі конфігурацію маршрутизатора змінити не можна, а список доступних команд можна подивитися ввівши такий символ «?»

Router> ?

Щоб перейти в привілейований режим, треба ввести команду «enable» в командний рядок. Після чого з'явиться нова підказка:

Router> enable

Router#

Для того, щоб вийти з привілейованого режиму потрібно просто ввести одне з цих слів: exit чи disable.

Щоб налаштувати маршрутизатор, режим конфігурації (configure terminal) повинен бути ввімкнений:

Router# configure terminal

Router(config)#

З режиму конфігурації є можливість перейти в інші підрежими серед яких:

- Режим конфігурації маршрутизатора

Для того, щоб його використати потрібно ввести команду router

Router(config-router)#

- Режим конфігурації інтерфейсу

Для входу в цей режим вводиться команда interface(тип)/(номер)

Router(config-if)#

- Режим конфігурації лінії

Для переходу в режим введіть команду line

Router(config-line)#

1.5 Способи захисту маршрутизатора

Рекомендовано фізично обмежувати доступ до мережевого обладнання, розмістивши його в закритій шафі або краще в окремій кімнаті. Однак паролі все ще залишаються основним засобом запобігання несанкціонованому доступу до мережевих пристроїв. На кожному приладі, навіть на домашньому маршрутизаторі, потрібно встановити паролі, щоб обмежити доступу. Пізніше буде розглянуто, як покращити безпеку, налаштувавши запит на ім’я

користувача разом з паролем. Наразі ми будемо використовувати тільки базові заходи для безпеки, використовуючи лише паролі.

Як вже згадувалося раніше, пристрої IOS використовують ієрархічні режими для забезпечення захисту. Для того, щоб підвищити рівень безпеки IOS краще створювати декілька паролів, кожен з яких належить до різних рівнів ієрархії.

Перелічені тут типи паролів включають:

Пароль привілейованого режиму — він обмежує доступ в привілейований режим.

Secret — це зашифрований пароль, який також обмежує доступ в привілейований режим, проте є більш надійним ніж попередній.

Пароль до консолі — обмежує доступ до приладів через підключення за допомогою консолі.

Пароль для ліній VTY - обмежує доступ до пристроїв через віддалений доступ(наприклад, telnet, rlogin, SSH).

Для кожного рівня доступу рекомендується використовувати свій особистий пароль аутентифікації. Хоча варто й визнати, що входити в систему з декількома різними паролями незручно, проте це необхідний захід для захисту інфраструктури мережі від несанкціонованого доступу.

Використовуйте команду **enable secret password** для захисту доступу до привілейованого режиму. Застарілою та менш безпечною версією цієї команди є enable password <password>. Хоча обидві команди підходять для налаштування аутентифікації перед входом у привілейований режим, все ж таки рекомендується використовувати enable secret. Оскільки пароль зашифрований, команда enable secret забезпечує вищий рівень безпеки.

Приклад команди для встановлення паролів:

R1(config)#enable secret class

Цей приклад показує, що для першого використання команди enable не потрібен пароль. Далі потрібно ввести команду enable secret class, і доступ в привілейований режим буде захищений. Варто звернути увагу, що з міркувань безпеки при введенні пароль не показується.

Консольний порт мережевих пристроїв повинен бути захищений принаймні надійним паролем. Це зменшує можливість неавторизованих працівників отримати доступ через кабель і таким способом ввійти в налаштування обладнання.

Щоб встановити пароль для рядка консолі в режимі глобальної конфігурації, потрібно ввести такі команди:

R1(config)#line console 0

R1(config-line)#password class

R1(config-line)#login

У режимі глобальної конфігурації команда «line console 0» використовується для того, щоб ввійти в режим конфігурації рядка для консолі. Нуль в прикладі використовується для позначення 1-го (а часто і єдиного) інтерфейсу консолі.

Наступна команда — password cisco вказує на пароль, який буде створено для консолі рядка.

А команда login налаштовує перемикач для аутентифікації при вході в систему. Якщо процес входу ввімкнено і пароль налаштовано, користувач консолі повинен буде ввести пароль для доступу до інтерфейсу командного рядка (CLI) [7].

Пароль для VTY

Канали VTY надають доступ до обладнання Cisco через протокол Telnet. За замовчуванням більшість маршрутизаторів компанії підтримують до 16 каналів VTY, що мають нумерацію від 0 до 15. Кількість підтримуваних на

маршрутизаторі Cisco каналів VTY залежить від типу маршрутизатора та версії IOS. Але зазвичай встановлюється п’ять каналів VTY, які за замовчуванням пронумеровані від 0 до 4, хоча за потреби ви можете налаштувати інші канали. Паролі повинні бути встановлені для всіх доступних каналів VTY. Та для всіх з'єднань є можливість встановлення лише одного паролю. При цьому зазвичай потрібно встановити унікальний пароль для каналу, щоб забезпечити адміністраторам резервний доступ, коли всі інші з'єднання зайняті.

Команди, які використовуються для того, щоб призначати паролі каналів VTY:

```
R1(config)#line vty 0 15
R1(config-line)#password class
R1(config-line)#login
```

За замовчуванням IOS має вбудовану команду входу на канали VTY. Це запобігає доступ через Telnet до пристрою без аутентифікації користувача. Якщо помилково була використана команда no login та через неї аутентифікація відключилася, то до мережі можуть приєднатися неавторизовані користувачі завдяки Telnet. Це становить певну загрозу безпеці маршрутизатора.

Під час перегляду файлу конфігурації існує ще одна важлива команда для захисту пароля. Це **service password-encryption**.

Вона шифрує паролі під час їх налаштування. Service password-encryption виконує шифрування тих, які є незахищеними. Воно застосовується лише до паролів, що знаходяться у файлі конфігурації, проте не використовується до тих, які відправлені через середовище передачі даних. Ця команда не дає можливість неавторизованим працівникам прочитати введену інформацію [8].

Якщо запустити команди show running-config або show startup-config перед виконанням команди шифрування, то незашифрований пароль буде можливо прочитати у вихідних даних конфігурації. Потім якщо використати команду service password-encryption, то після цього паролі будуть зашифровані й цю дію неможливо буде скасувати.

Хоча паролі й можуть захистити мережу від неавторизованих користувачів, все одно краще використовувати повідомлення про те, що до пристрою може отримати доступ лише той, хто був авторизований. Якщо користувача звинувачують у несанкціонованому доступі, такі **банери** можуть бути корисними під час судового розгляду. Деякі судові системи забороняють переслідувати або відстежувати дії осіб без попередження. Точний зміст або формулювання банера залежать від корпоративної політики компанії та місцевого законодавства певної країни чи міста. Нижче наведено можливі тексти формулювань у таких інформаційних банерах:

«Лише авторизовані користувачі можуть отримати доступ до пристрою».

«Ваші дії можуть відстежуватися».

«Будь-яке несанкціоноване використання буде переслідуватися законом».

Оскільки кожен, хто намагається отримати доступ до пристрою, може побачити банер, то формулювання повідомлення має бути обережним. Не слід використовувати у своїх повідомленнях фрази типу "Добрий день". Якщо користувач порушує функціонування мережі, то після незаконного вторгнення, важко довести злочин якщо при вході в систему було лише привітання.

Хоч вигадати потрібний банер не важко, проте його текст необхідно ретельно продумати. Він не повинен давати запрошення для отримання доступу до пристрою для кожного користувача. У ньому має бути зазначено, що доступ можуть мати лише ті, хто є авторизованими. Крім того, банер може містити графік вимкнення мережі та іншу інформацію, яка може бути корисною для інших користувачів системи. IOS пропонує велику кількість типів банерів. Повідомлення поточного дня — доволі поширений серед них. Він часто використовується як банер, через те, що його бачать всі приєднані термінали.

Для використання команди banner motd, потрібно пам’ятати про роздільник, щоб можна було визначити вміст банерного повідомлення. За словами banner motd слідують пробіл та символ, який виступає у ролі

роздільника. Потім вводиться один чи більше рядків тексту для створення банерного повідомлення. Другий розділовий символ означає його кінець. Розділювачем може бути будь-який символ, якого немає в цьому повідомленні. Тому досить часто використовуються такі символи, як «#» через їх рідке розташування в тексті.

Використовуйте такий синтаксис, щоб налаштувати повідомлення в режимі глобальної конфігурації:

R1(config)#banner motd# your message #

Після виконання команди банер показується у всіх наступних спробах отримати доступ до пристрою, до тих пір, поки його не буде видалено іншою командою.

Політику блокування облікового запису користувача можна застосувати на пристроях Cisco з метою запобігання деяким видам атак. Нижче буде пояснено, як заблокувати користувачів за допомогою команди **login block-for**, у випадку, коли зареєстрована особа перевищує певну кількість неправильних спроб для входу. Якщо ви введете неправильні облікові дані певну кількість разів, то команда входу заблокує всі з'єднання telnet та SSH до цього маршрутизатора.

Синтаксис команди:

Заблокувати на <Час у секундах> <Максимальна кількість спроб > протягом <Час у секундах>

Команду слід ввести в режимі конфігурації. Ось приклад:

R1(config) #login block-for 80 attempts 3 within 15

Наведена вище команда блокує всі підключення до маршрутизатора R3 на 80 секунд, якщо облікові дані вводяться 3 рази неправильно в межах 15 секунд. Якщо ви порушите це правило, ви отримаєте саме таке повідомлення в терміналі консолі.

Функція **No Service Password-Recovery**

У версії 12.3 програмному забезпеченні Cisco IOS та більш пізніх ця функція не дозволяє користувачам, які мають доступ до пристрою через консоль скидати встановлені паролі. Вона також не дасть можливість зловмиснику змінювати значення конфігураційного реєстру та звертатися до NVRAM.

>no service password-recovery

Програмне забезпечення Cisco IOS передбачає спосіб відновлення пароля шляхом переходу в режим монітора ПЗУ після натискання клавіші «Break» в момент запуску системи. Після цього програмне забезпечення пристрою може бути перезавантажене в новій конфігурації системи з іншим паролем.

Завдяки процесу відновлення поточного пароля будь-який користувач, який має консольний доступ може звернутися до пристрою і його мережі. Але функція No Service Password-Recovery запобігає виконанню послідовності після натискання клавіші Break та входу під час запуску системи.

Якщо пристрій перебуває в режимі, коли пароль неможливо відновити, рекомендується зберегти копію конфігурації маршрутизатора та застосувати її резервне копіювання. Якщо вам потрібно відновити пароль пристрою Cisco IOS, то вся конфігурація буде видалена після ввімкнення цієї функції.

Тайм-аут EXEC

Щоб вказати інтервал часу для інтерпретатора команд EXEC, який повинен очікувати введення користувачем перед закінченням сеансу, потрібно ввести команду налаштування exec-timeout у командному рядку. Вона використовується для завершення сеансу в неактивних каналах vty та tty. За замовчуванням після 10 хвилин бездіяльності сеанс буде вимкнено.

```
line con 0
  exec-timeout <Час у хвилинах>
line vty 0 4
  exec-timeout <Час у хвилинах>
```

Повідомлення про порогові значення пам'яті

Функція сповіщень про порогові значення пам'яті, яка з'явилася в програмному забезпеченні Cisco IOS, допомагає зменшити дефіцит пам'яті на пристрої. Ця функція використовує два методи для досягнення цілі: Повідомлення про порогові значення та резервування пам'яті.

Повідомлення про порогові значення пам'яті записується в журнал подій та вказує, що обсяг доступної пам'яті на пристрої нижче вказаного порогового значення. Наступний приклад конфігурації показує, як активувати цю функцію командою глобальної конфігурації memory free low-watermark. Таким чином, коли обсяг доступної пам'яті нижчий за вказане граничне значення, і коли обсяг перевищує вказаний ліміт на 5%, пристрій може встановити з'єднання з повідомленнями.

```
memory free low-watermark processor <значення порогу>
memory free low-watermark io <значення порогу>
```

Резервування пам'яті використовується для забезпечення достатньої її кількості для найбільш важливих повідомлень. Наступний приклад конфігурації демонструє, як увімкнути цю функцію. І навіть якщо пам’ять пристрою вичерпується, то гарантується, що процес управління продовжить працювати.

```
memory reserve critical <значення>
```

Служба Finger

Маршрутизатори Cisco мають служба finger, яка допомагає дізнатися, які користувачі в певний момент часу виконали реєстрацію на приладі. Проте в більшості випадків ця інформація є конфіденційною. Саме тому часто вона може виявитися корисною злодіям. Цю службу можна відключити за допомогою команди no service finger.

Малі служби TCP і UDP

За замовчуванням усі версії Cisco IOS починаючи з версії 11.3 мають такі служби як:

- chargen
- echo
- discard

Всі вони, а особливо служби протоколу UDP, досить рідко використовуються в роботі цілеспрямовано. Однак вони можуть бути застосовані для того, щоб проводити DoS, а також ще для деяких видів атак, які в інших випадках можна запобігти завдяки фільтрації пакетів.

І таким способом хакери можуть надсилати DNS-пакети з підробленою вихідною адресою, щоб стати DNS-сервером, який в інших випадках буде недоступним, та підробленим вихідним портом служби. У випадку, коли такий пакет направляється на порт UDP, результатом може стати пакет DNS, який буде надісланий на вказаний підроблений сервер. Хоча списки доступу можуть допомогти зменшити вплив та ризик несанкціонованого використання цих служб або зробити його більш безпечним, проте у багатьох випадках краще вимкнути їх на всіх маршрутизаторах через нечасте використання. З IOS 12.0 Cisco автоматично вимикає ці служби. Але якщо ви використовуєте версію нижче, то рекомендується оновити її, а в випадку, коли це неможливо, відключити їх за допомогою команди no service tcp-small-servers і no service udp-small-servers [9].

Протокол NTP

Використання NTP або протоколу мережевого часу не повинна приносити ніяких ризиків, але велика кількість надлишкових служб може стати причиною знаходження зловмисниками можливостей для заволодіння особистих даних. Під час використання NTP дуже важливо вказувати надійне джерело синхронізації даних та використовувати відповідний механізм аутентифікації. Порушення коректної роботи цього протоколу може спричинити проблеми в функціонуванні деяких протоколів безпеки маршрутизатора. Тому якщо інтерфейс маршрутизатора не використовує NTP, ви можете скористатися командою ntp disable, щоб його вимкнути.

Протокол CDP

Протокол виявлення Cisco (Cisco Discovery Protocol) використовується для деяких функцій управління мережею. Хоча використання CDP є небезпечним, оскільки він дозволяє всім системам безпосередньо підключеного сегмента мережі виявити, що маршрутизатор є пристроєм Cisco. Це призводить до заволодіння інформацією про версію та номер моделі Cisco IOS, яка використовується приладом. І дані можуть бути використані при плануванні атак на роутер. Інформація протоколу доступна лише для безпосередньо підключених систем. Його можливо відключити, скориставшись командою для налаштування no cdp running. Також цей протокол можна відключити на вказаному інтерфейсі за допомогою команди no cdp enable.

Боротьба з підміною адрес за допомогою списків доступу

Встановлення списку доступу багато в чому залежить від умов роботи певної мережі. Однак головне завдання полягає у відсіюванні пакетів, які надходять на ті інтерфейси, маршрут до яких не є допустимим. Можна привести приклад того, коли на маршрутизатор, який надає доступ в мережу Інтернет приходить інформація з вихідною адресою пристрою, який знаходиться поза цією мережею й такий пакет повинен бути відкинутий.

Якщо це дозволяють ресурси приладів, то такий спосіб протидії від підмін слід застосовувати на всіх інтерфейсах, на яких можливо визначити, що трафік, який приходить є правомірним. Хоча провайдери передачі даних використовують певні методи для фільтрації потоку вхідної інформації, вони мають обмежені можливості налаштування вхідних списків доступу.

Як правило, фільтри повинні використовуватися разом із вхідними списками доступу. Іншими словами це означає, що пакети повинні фільтруватися на тих інтерфейсах, через які вони надходять на сам маршрутизатор, а не на тих, через які вони виходять з нього. Фільтрація налаштовується з використанням команди конфігурації ip access-group list in. Якщо є списки для перевірки справжності адрес, то вони завжди повинні відхиляти пакети даних з широкомовними адресами джерела або із зарезервованим адресами зворотного зв'язку. Також списки доступу повинні фільтрувати всі перенаправлення ICMP, незалежно від того, хто є відправником або отримувачем.

Також варто розповісти про такий протокол, як **SSH**. Це мережевий протокол прикладного рівня, який призначений для безпечного віддаленого доступу та використовується для обміну інформацією між пристроями через зашифрований канал. Даний протокол ефективний тим, що шифрує всю інформацію, а не окремі його фрагменти, що передаються мережею, на відміну від протоколу telnet. В основному він використовується для віддаленого управління даними користувача на сервері, запуску утиліт команд та для забезпечення роботи з базами даних.

Важливою особливістю SSH вважають його розширені можливості, наприклад, його здатність передавати будь-який мережевий протокол через виділений шифрований канал зв'язку. Це дає людині, яка ним користується, засіб віддалено виконувати роботу на власному комп’ютері та надсилати різні види інформації(текст, відео, аудіо, графічні матеріали) по захищеному шифрованому каналу.

Для того, щоб завантажувати передану інформацію зручніше та швидше, протокол також має функцію стиснення даних, яка є можливою до виконання перед процедурою самого шифрування.

Протокол SSH включає має три рівні:

Перший є протоколом транспортного рівня, який використовується для аутентифікації сервера під час включення системи та для забезпечення цілісності та захищеності інформації. Другий є протоколом аутентифікації користувача та призначений для того, щоб забезпечити перевірку для сервера при його першому запиті. Третій є протоколом з'єднання, який використовується для того, щоб забезпечити роботу логічних каналів поверх одного SSH-з'єднання.

Підключення до віддаленого комп'ютера через протокол відбувається за допомогою командного інтерпретатора. Під час сесії всі команди, введені на вашому локальному пристрої будуть відправлятися через зашифрований тунель і виконуватися на віддаленому сервері. Для цього на ній повинно бути запущено спеціальне програмне забезпечення, яке перевіряє користувача та надає йому доступ при успішній авторизації [10, 11].

Для роботи SSH-протоколу на локальному комп'ютері повинен бути встановлений SSH-клієнт, який має інформацію для успішної аутентифікації та авторизації. Клієнт SSH встановлено в більшості дистрибутивів Linux й інших UNIX-подібних систем, а для Windows потрібно встановити спеціальну програму. Як приклад, серед найбільш популярних можемо виділити Putty.

1.6 Постановка задачі

Після проведення аналізу літератури, можна сформулювати таку мету даної роботи: потрібно розробити веб-орієнтований графічний інтерфейс, який буде дозволяти автоматично налаштовувати різні команди для захисту, аутентифікації маршрутизатора. Дана система повинна мати можливість швидкого копіювання необхідних даних та перенесення їх в будь-який симулятор чи на реально обладнання мережі Cisco. Інтерфейс має допомогти зменшити час на початкове налаштування. Також він повинен мати простий інтерфейс, який буде зрозумілий навіть людині, яка не має спеціальних знань в області телекомунікацій.

Для того, щоб створити графічний інтерфейс, необхідно розробити веб-сторінку, на якій користувач зможе вводити свої дані та отримати перелік команд для внесення їх в маршрутизатор. Для реалізації системи потрібно спочатку змоделювати мережу з використанням маршрутизаторів Cisco в симуляторі Packet Tracer, розробити веб-сайт для налаштування параметрів безпеки, а потім перевірити працездатність отриманих результатів.

2 МОДЕЛЮВАННЯ МЕРЕЖІ ТА РОЗРОБКА ВЕБ-СТОРІНКИ ІНТЕРФЕЙСУ

2.1 Конфігурація мережі з використанням симулятору Cisco Packet Tracer

Симулятори для моделювання мереж активно використовуються інженерами та адміністраторами для моніторингу, проектування та повного аналізу комп'ютерних систем. Вони надають можливість перевіряти роботу певного обладнання та викреслюють можливість спричинення небезпеки та виникнення технічних несправностей. Серед найбільш популярних та функціональних можемо виділити такі симулятори, як EVE-NG, VIRL, GNS3, OPNET. Вони можуть використовуватися для проектування мереж різної складності, та мають в собі інструменти для навчання та вивчення багатьох видів обладнання та реалізації логіки їх взаємодії між собою.

У першій частині роботи було коротко описано компанію Cisco Systems та визначено, що вона є одним із лідерів у сфері телекомунікацій. Серед програмних продуктів звернемо увагу на Cisco Packet Tracer. Це емулятор, який дозволяє користувачам проектувати та перевіряти роботу різноманітних мереж з підтримкою великої кількості протоколів.

Під час роботи в симуляторі є можливість побачити весь процес передачі інформації в мережі, переглянути всі можливі налаштування пристроїв та побудувати систему, яка буде складатися з досить великої кількості різного обладнання, реальну роботу якого програма може імітувати. Серед них:

- комутатори
- маршрутизатори
- хаби
- бездротові девайси
- сервери
- телефони

- телевізори
- сплітери
- принтери
- планшети
- прилади для розумного дому

Перевагами Cisco Packet Tracer є зрозумілий інтерфейс для початківців, можливість роботи в програмі на різних системах, надзвичайно широка база обладнання, вигляд якого також можливо переглянути, наявність великої кількості інформації про налаштування систем в симуляторі будь-якої складності, постійна підтримка та додавання нових функцій [12].

На Рис. 2.1 зображений інтерфейс програми.

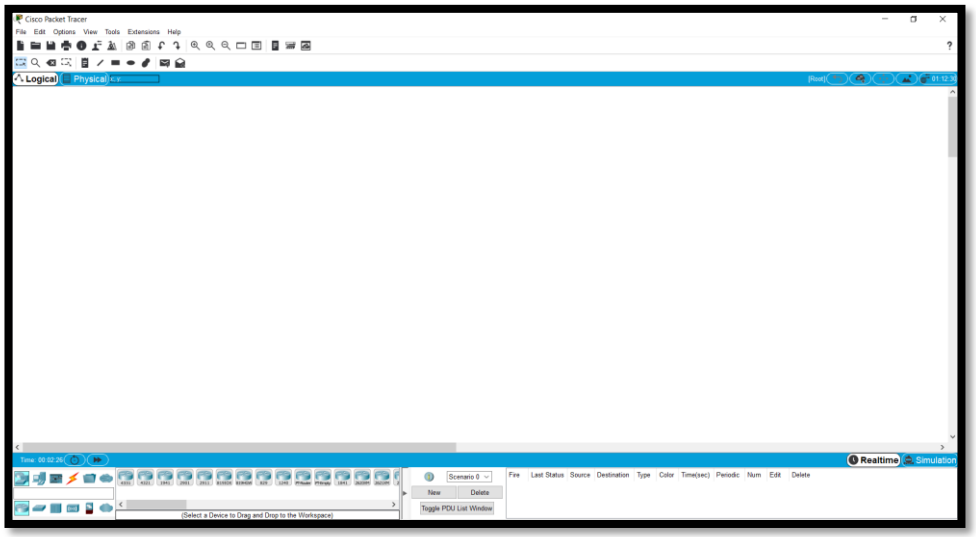

Рис. 2.1 – Інтерфейс Cisco Packet Tracer

За допомогою симулятору створено модель мережі для кращого розуміння інтерфейсу(Рис. 2.2). Оскільки головною задачею є можливість налаштування маршрутизатора, то на схемі показано мережу з декількома роутерами, на яких необхідно ввести команди. Для моделювання використано модель Cisco 2811 через те, що він часто використовується в корпоративних мережах.

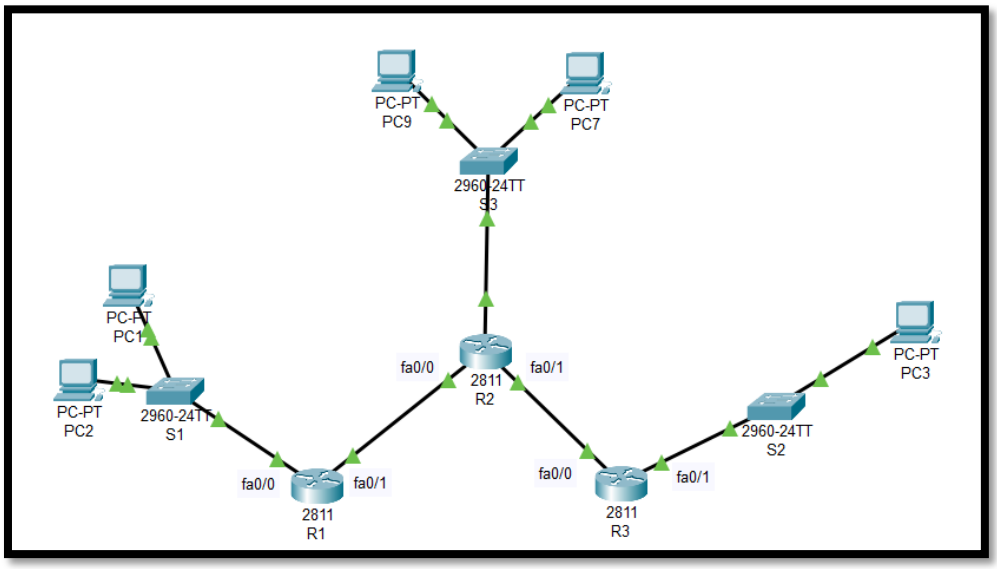

Рис. 2.2 Моделювання мережі

Для налаштування маршрутизатора використовуються різні типи команд. Спочатку використовуються команди, які визначають інтерфейс(зазвичай їх декілька, вони слугують для з'єднання з іншими мережами), унікальну назву, ip-адресу та маску підмережі.

>enable

>configure terminal

>hostname R1

>interface fa0/0

>ip address 192.168.0.1 255.255.255.0

>enable secret passsecret32

Далі виконується налаштування різних режимів роботи та паролі до них та кількість часу, яка необхідна для користування кожного з режимів. Команди виглядають так:

```
>line console 0
>password consolepass34
>exec-timeout 40 0
>logging synchronous
>login
>exit

>line vty 0 4
>transport input ssh
>exec-timeout 40 0
>logging synchronous
>exit


>line aux 0
>password cisco
>exec-timeout 40 0
>login
>logging synchronous
>exit
```

Також необхідні для введення команди, які забезпечують шифрування паролів, визначають їх мінімальну довжину та повідомляють користувача про вхід в певний режим.

>service password-encryption

>security passwords min-length 10

>no ip domain-lookup

>login block-for 100 attempts 3 within 60

>banner motd #This is a private system. You must be logged in to use this system.#

Для налаштування входу через SSH використовуємо:

>ip domain-name ssh.com

>crypto key generate rsa

>username admin privilege 15 secret admin23admin14

2.2 Розробка веб-сторінки інтерфейсу з використанням мови JavaScript

Для реалізації інтерфейсу було прийнято рішення використовувати мову JavaScript. Це високорівнева мова для програмування сценаріїв, яка має такі можливості:

- реалізація інтерактивної роботи сторінки
- додавання анімацій та форми поведінки
- перевірка на правильність введених значень в формах
- заміна стилів для елемента або всього сайту
- запам'ятовування інформації про користувача в браузері
- створення різних віджетів(наприклад, меню, яке розкривається)

JavaScript належить до мов з динамічною типізацією. У число основних особливостей JS входять:

- Інтерпретована мова – це означає, що код додатку інтерпретується

під час звернення і попередня компіляція не потрібна.

- Функції в JavaScript можна повертати з інших функцій, передавати їх як параметри в інші функції та привласнювати іншим змінним.
- Динамічна типізація — тип даних визначається в момент надання значення константі або змінній.
- Підтримка прототипного та об'єктно-орієнтованого підходу.
- Універсальність — всі популярні браузери підтримують JavaScript.

Важлива особливість JavaScript — це те, що його інфраструктура досягла значного рівню розвитку. Навколо цієї мови програмування сформовано численне товариство. Розробникам доступні потужні інструменти, наприклад:

- Допоміжні бібліотеки (Ramda, Lodash, Underscore, Moment).
- Бібліотеки та фреймворки для створення додатків (jQuery, Vue, Angular, React).
- Збирачі (Gulp, Webpack).
- Генератори статичних сайтів (React static, Gatsby, Cuttlebelle, Next.js).

В першу чергу JavaScript використовується під час front-end розробки. Він разом з CSS та HTML завжди входить в список інструментів для створення сайтів. Також завдяки JavaScript створюються різні додатки, які розгортуються на стороні клієнта в його браузері. Вони забезпечують інтерактивність сайтів. Наприклад, у випадку, коли користувач сторінки виконує заповнення однієї з присутніх форм на сайті та після цього натискає певну кнопку, то відповідь на цю дію забезпечується кодом, який написаний на JavaScript [13].

Сфери застосування JavaScript не обмежуються браузерами та веб-додатками. За допомогою цієї мови вирішують такі завдання:

- Розробка нативних додатків. Наприклад, за допомогою фреймворку React Native створюються додатки для Android і iOS.

- Серверна розробка. Так Node.js використовується для бекенд-розробки.
- Програмування обладнання та побутової техніки, як приклад, платіжних терміналів або ж телевізійних приставок.
- Розробка десктопних додатків. JavaScript часто може бути застосований в офісних пакетах Microsoft та OpenOffice, а також в додатках компанії Adobe.

Також варто розповісти про один з найбільш популярних та використовуваних бібліотек JavaScript.

jQuery — це бібліотека JavaScript, яка була розроблена Джоном Ресігом у 2006 році. Вона має відкритий код та містить в собі уже готові функції, а необхідні операції виконуються з використанням коду JavaScript. Бібліотека значно спрощує написання коду та взаємодію між елементами веб-сторінки. В основному вона використовується для управління та переміщенню по HTML документах, створення спеціальних ефектів анімації та обробки подій елементів сторінки. jQuery - це легка, швидка та багатофункціональна бібліотека. Її головна перевага – це простота та стисліть коду.

Для порівняння візьмемо такий код на JavaScript:

```
var p = document.getElementsByClassName("paragr");
for (var i = 0; i <  p.length; i++) {
p[i].style.color = "red";
}
```

Для виконання таких же дій за допомогою бібліотеки необхідно прописати лише один рядок:

```
$(" paragr ").css("color", "red");
```

З даного прикладу витікають такі переваги JQuery:

- Швидкість написання коду
- Швидкість засвоєння
- Швидкість роботи

Також дана бібліотека є кросбраузерною, тобто її підтримують всі види сучасних браузерів. У мережі Інтернет є велика кількість готових прикладів коду написаних на JQuery, що дає можливість їх використовувати та не витрачати час на створення багатьох шаблонів. Також усі версії бібліотеки повністю поєднуються між собою, що дає можливість використовувати її навіть на застарілих пристроях.

При створенні інтерфейсу також були використані HTML та CSS.

Hypertext Markup Language(HTML) — це стандартна мова розмітки гіпертексту, яка використовується для зображення документів у веб-браузері. HTML використовує теги для позначення частин вмісту як різних частин веб-документа, таких як абзаци, заголовки та посилання [14].

Вперше мова була розроблена Тімом Бернерсом Лі в 1990 році під час роботи в Європейській організації з ядерних досліджень. HTML був однією з ключових інноваційних технологій, що використовувалися для публікації першого у світі веб-сайту. З того часу HTML значно оновився та розширився, але його основне призначення — форматувати та структурувати веб-сторінки залишається незмінним [15, 16].

HTML використовує заздалегідь визначені теги та елементи, які вказують браузеру, як правильно зображають вміст.

Структура сторінки HTML: вона складається з основних елементів, які можна порівняти з будівельними блоками, на основі яких створюються всі веб-сторінки. На Рис. 2.3 показана найпростіша форма документу HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Заголовок документа</title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <p>Текст</p>
  </body>
</html>
```

Рис. 2.3 – Структура HTML

Основні теги такі:

<html> — вказує програмі перегляду сторінок, що це документ HTML.

<title> — передає назву програмі перегляду сторінок.

<head> — передає інформацію, яка не розміщається в тілі документу.

<body> — вказує тіло документа, яке показується на веб-сторінці.

CSS — це мова, яка надає можливість додавати в HTML-документи дані про їх форматування. Наприклад, такі параметри, як кольори, відступи, шрифти, позиціювання. CSS досить проста у своїй структурі, тому зрозуміти її може будь-яка людина, яка має бажання створювати веб-сторінки. Код CSS діє як таблиця стилів. Він визначає розміщення всіх об'єктів, колір, типографію та надає спосіб змінювати зовнішній вид сайтів.

CSS визначає, як повинен виглядати елемент на веб-сторінці. Стилі можна застосовувати за допомогою CSS у самому документі HTML, або в окремому файлі для зручності.

Використовуючи CSS, розробники пишуть правила стилю, які повідомляють сайту, як певний елемент HTML повинен зображатися на сторінці. Наприклад, одне правило стилю може визначати висоту та ширину потрібного зображення, а інше може встановити розмір тексту на цьому веб-сайті.

Мова CSS пропонує широкий спектр властивостей, які використовуються для застосування стилів до різних елементів веб-сторінки. Ці ознаки стосуються розміру елемента, його кольору, його меж, де він з’являється на веб-сторінці тощо. CSS має нескладний синтаксис. Кожен зі стилів складається з певного списку селекторів, які мають певні властивості та значення розміщенні в блоці визначення. Нижче показано приклад селектору, який задає правила для оформлення абзаців [16].

```
p {
  color: white;
  font-size: 500px;
  border: 2px solid black;
 }
```

Також при технічній реалізації веб-інтерфейсу був використаний скрипт clipboard.js.

Це сучасний плагін, який дозволяє користувачам значно простіше реалізовувати в себе на сайті функцію копіювання даних в буфер обміну, за допомогою натиснення певної кнопки, яка зображена на самій веб-сторінці. Даний скрипт ґрунтується на роботі execCommand(метод для роботи з інформацією в певній частині сайту) та API Selection(забезпечує можливість для читання та обробки конкретного діапазону тексту).

Сценарій можна використовувати у всіх найбільш часто використовуваних браузерах (Google Chrome, Mozill Firefox, Safari, Internet Explorer, Opera)

Його переваги такі:

- Він є незалежним, тобто для того, щоб працювати з clipboard.js не потрібно ніяких сторонніх бібліотек
- Досить компактний розмір
- При роботі не потрібен Flash

Перелічимо основні пункти для налаштування роботи плагіну clipboard.js на нашому веб-сайті:

Перш за все необхідно з офіційного сайту завантажити архів з файлами. А потім виконати його розпаковування в потрібний каталог.

Після цього варто під'єднати файл скрипту в наш HTML-документ за допомогою команди:

<script src = "../clipboard.min.js"> </ script>

І в цьому ж документі обрати кнопку після натиснення якої буде відбуватися копіювання та сам елемент, звідки буде скопійована вся інформація.

Команда така:

<input type="button" value="Скопіювати" class="copy_button" data-clipboard-target=".comands">

А для початку роботи, потрібно виконати ініціалізацію функції Clipboard в нашому файлі з JavaScript:

new ClipboardJS('.copy_button');

Після того, як всі дію будуть зроблені завдяки скрипту з’являється можливість після натиснення кнопки «Скопіювати» отримувати дані з блоку comands в буфер обміну та використовувати їх для налаштування потрібних маршрутизаторів [17].

Під час створення файлу головного файлу для реалізації інтерфейсу(script.js) використано такі функції:

ClipboardJS() – копіювання даних для налаштування роутеру

hideComands() – приховування блоку з командами до моменту правильно введеної інформації

showComands() – показ блоку з командами

validateIp(ip) – валідація введеної IP-адреси пристрою

validateMask(mask) – валідація введеної маски підмережі

validateForm() – валідація всіх даних, які вводяться в форму

generate() – генерація команд з підставленням введеної в текстові поля

інформації.

Для застосування валідації маски використано масив, в якому записані всі можливі її варіанти для введення.

```
var mask_array = ['255.255.255.255',
                  '255.255.255.254',
                  '255.255.255.252',
                  '255.255.255.248',
                  '255.255.255.240',
                  '255.255.255.224',
                  '255.255.255.192',
                  '255.255.255.128',
                  '255.255.255.0',
                  '255.255.254.0',
                  '255.255.252.0',
                  '255.255.248.0',
                  '255.255.240.0',
                  '255.255.224.0',
                  '255.255.192.0',
                  '255.255.128.0',
                  '255.255.0.0',
                  '255.254.0.0',
                  '255.252.0.0',
                  '255.248.0.0',
                  '255.240.0.0',
                  '255.224.0.0',
                  '255.192.0.0',
                  '255.128.0.0',
                  '255.0.0.0',
                  '254.0.0.0',
                  '252.0.0.0',
                  '248.0.0.0',
                  '240.0.0.0',
                  '224.0.0.0',
                  '192.0.0.0',
                  '128.0.0.0',
                  '0.0.0.0'];
```

Рис. 2.4 – Можливі маски підмережі

А для того, щоб введена ip-адреса була в діапазоні тих, які існують, створений регулярний вираз, зображений на Рис. 2.5.

```
var format_ip = /^([01]?[0-9][0-9]?|2[0-4]
[0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4]
[0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4]
[0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4]
[0-9]|25[0-5])$/;
```

Рис. 2.5 – Регулярний вираз для ip-адреси

Для перевірки введених даних у формі також використані регулярні вирази, які перевіряють правильність інформації кожного поля. Функція, яка забезпечує валідацію форми також додає певні класи до елементів, в залежності від того, чи коректно все було заповнено.

```
//Функція валідації форми
function validateForm() {
    var name_router = $('#name_router').val();
    var num_int = $('#num_int').val();
    var priv_mod = $('#priv_mod').val();
    var ssh_mod = $('#ssh_mod').val();
    var console_mod = $('#console_mod').val();
    var name_pattern = /^[a-zA-z0-9]{1,}$/;
    var int_pattern = /^[a-zA-z]{1,3}[0-9]{1,3}\/[0-9]{1,3}$/;
    var password_pattern = /^[a-zA-z0-9@#$%^&*\.\-_]{10,}$/;
    var valid = 0;
    if (name_router.match(name_pattern)) {
        $('#name_router').removeClass("error");
        $('#name_router').addClass("correct");
    } else {
        valid++;
        $('#name_router').removeClass("correct");
        $('#name_router').addClass("error");
    }
```

Рис. 2.6 – Перевірка форми

Для того, щоб реалізувати веб-сторінку інтерфейсу необхідне інтегроване середовище розробки(IDE). На цей момент існує досить велика кількість різноманітних програм, які допомагають спростити написання коду. З найбільш популярних можна виділити Sublime Text, Visual Studio Code, Notepad++, Atom, Vim. Кожен з них має свої переваги та недоліки(швидкість роботи, можливість роботи на різних пристроях чи зі складними проектами). Для роботи було обрано редактор Brackets. Він був створений відомою компанією Adobe. Дане середовище розробки має такі переваги:

- кросплатформеність
- зручний інтерфейс
- програма є безкоштовною
- наявність плагінів
- підтримка синтаксису різних мов

3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕРФЕЙСУ ДЛЯ НАЛАШТУВАННЯ ПАРАМЕТРІВ БЕЗПЕКИ МАРШРУТИЗАТОРА

3.1 Веб-інтерфейс для налаштування безпеки та аутентифікації маршрутизаторів

За допомогою програми Cisco Packet Tracer було виконано конфігурацію мережі. У симуляторі зроблено налаштування маршрутизатора, виконано ідентифікацію з введенням адреси, назви та маски, зазначено паролі до різних режимів роботи та занесено команди, які покращують безпеку та захищеність. Визначено, що для виконання даних налаштувань необхідна велика кількість часу. Якщо кількість приладів є значною, то виконання рутинних дій зменшує ефективність праці та забирає час, який можна витратити на інші види робіт. Через це досить ефективним є використання графічного інтерфейсу для налаштування параметрів безпеки та аутентифікації користувача.

Веб-інтерфейс реалізований з використанням JavaScript, HTML та CSS. Його дизайн та сам вигляд є досить зрозумілим навіть для початківців. Інтуїтивно можна зрозуміти, яка частина сторінки за що відповідає.

На Рис. 3.1 можна побачити вигляд графічного інтерфейсу після відкриття сайту.

Рис. 3.1 – Інтерфейс веб-сторінки

Після відкриття сторінки з графічним інтерфейсом можемо побачити веб-форми для введення даних. Спочатку маємо можливість занесення даних про IP та маску підмережі маршрутизатора. Також спостерігаємо форму для введення різних параметрів серед яких є ім’я, номер інтерфейсу та паролі для привілейованого режиму, входу через консоль та SSH. Після введення всіх необхідних даних у нас є можливість використання 2 кнопок. «Згенерувати» відповідає за коректне внесення введених даних в блок з командами. «Скопіювати» відповідає за функцію копіювання всіх згенерованих команд.

Графічний інтерфейс також має валідацію введених даних. Так при некоректному введенні поле, значення якого не відповідає певному формату буде підсвічено червоним кольором, що буде сигналізувати про те, що дані потрібно змінити. Також буде відсутня можливість переглянути згенеровані команди через помилку.

Реалізована валідація має певні принципи роботи. Так, якщо введена IP-адреса складається не з чисел, або ж виходить за допустимі рамки, то можливість переглянути створений код буде відсутня. Також паролі повинні мати розмір більше ніж 9 символів для того, щоб підібрати їх було складніше та ризик того, що їх зможуть підібрати став нижче. Якщо ім’я роутеру або його номер інтерфейсу матиме некоректний формат, то в такому випадку на веб-сторінці буде зрозуміло показано про наявність помилкових введених даних.

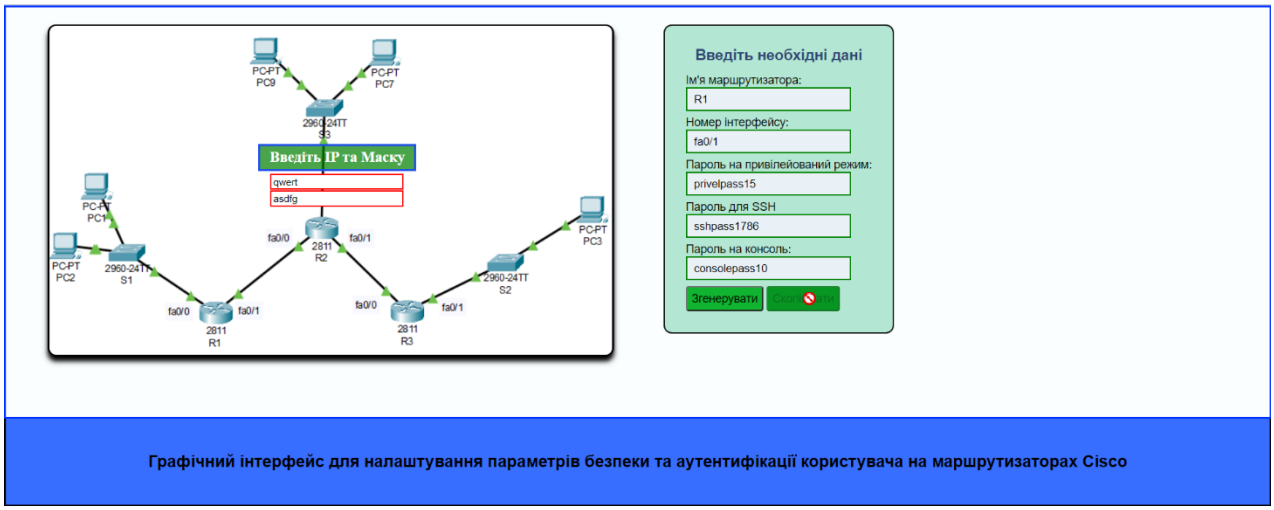

Рис. 3.2 – Невірний формат IP-адреси та маски

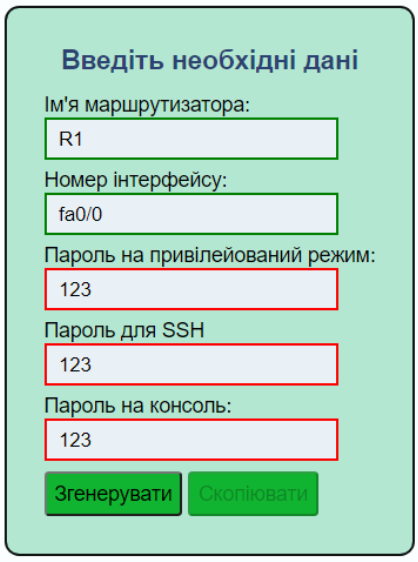

Рис. 3.3 – Недостатня кількість символів для паролів

Для користування графічним інтерфейсом мережевому адміністратору потрібно ввести правильно всі дані та натиснути на кнопку «Згенерувати». Після того, як це буде зроблено праворуч від інших блоків з’явиться текст команд, який можна скопіювати в буфер обміну. На Рис. 3.4 зображено генерацію необхідних команд після введення всієї інформації(якщо все виконано правильно, то кнопка «Скопіювати» повинна розблокуватися).

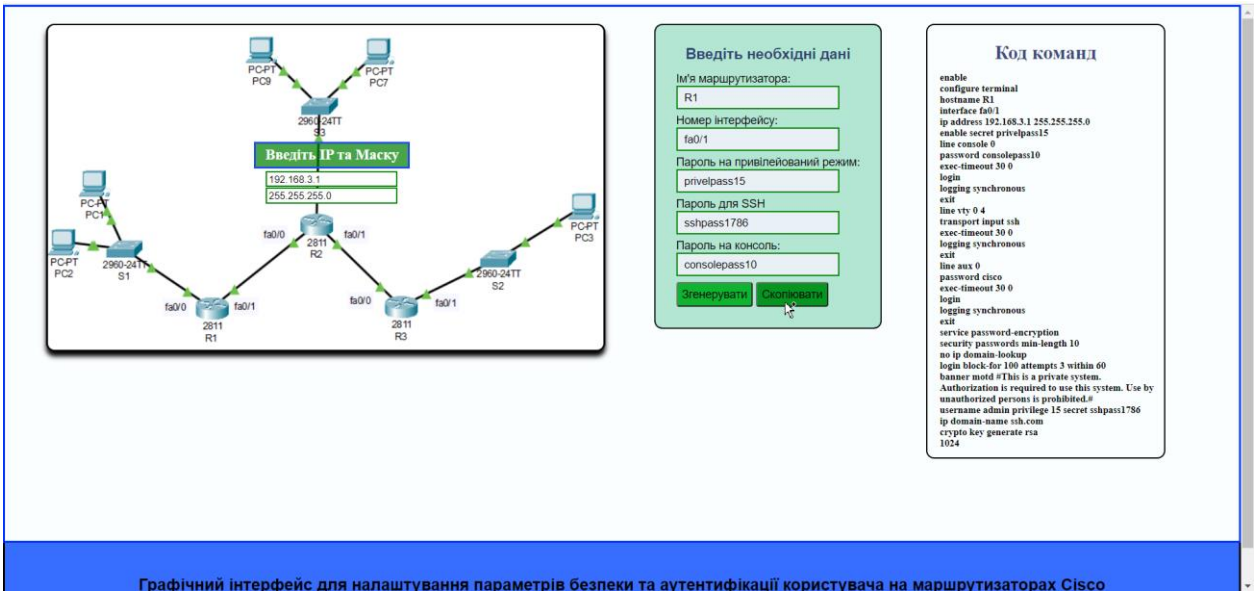

Рис. 3.4 – Генерація блоку з командами для маршрутизатора

На Рис. 3.5 приведено приклад роботи кнопки «Скопіювати». При наведенні на неї курсор показує можливість копіювання та після її натиснення виділяється весь текст в блоці з командами та зберігається в буфері обміну пристрою.

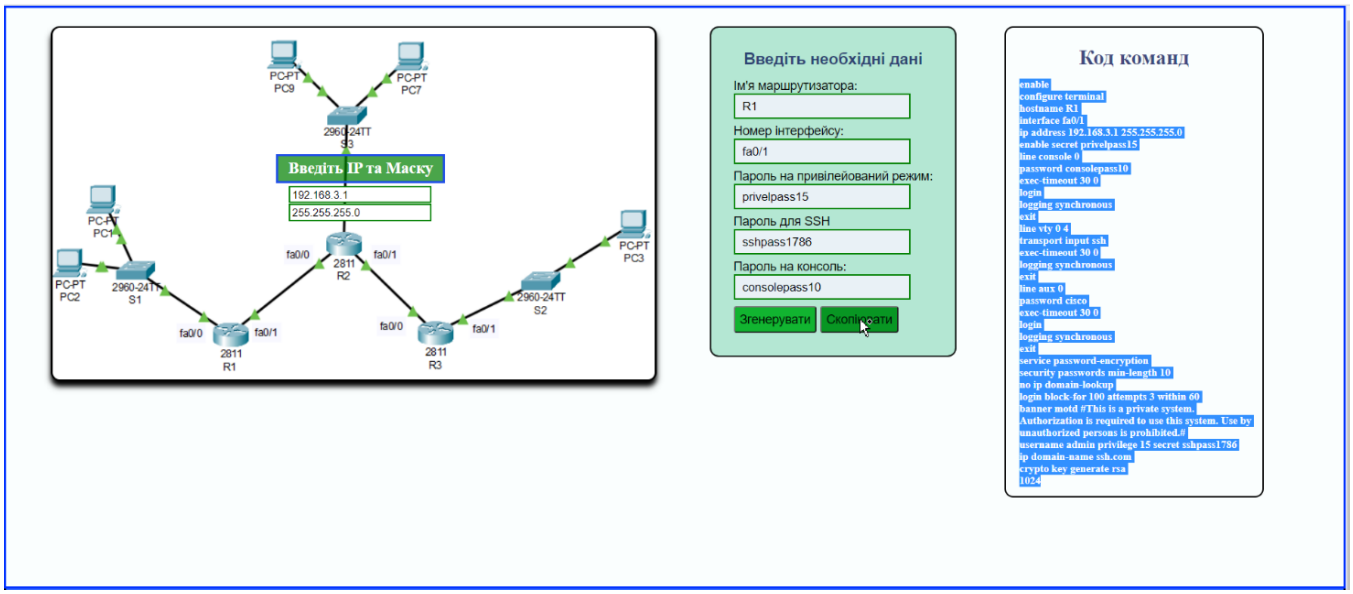

Рис. 3.5 – Копіювання команд

3.2 Тестування графічного інтерфейсу в симуляторі Cisco Packet Tracer

Для того, щоб перевірити правильність роботи створеного графічного інтерфейсу потрібно виконати тестування в програмі Cisco Packet Tracer. Для цього введемо всі дані для роутера R1.

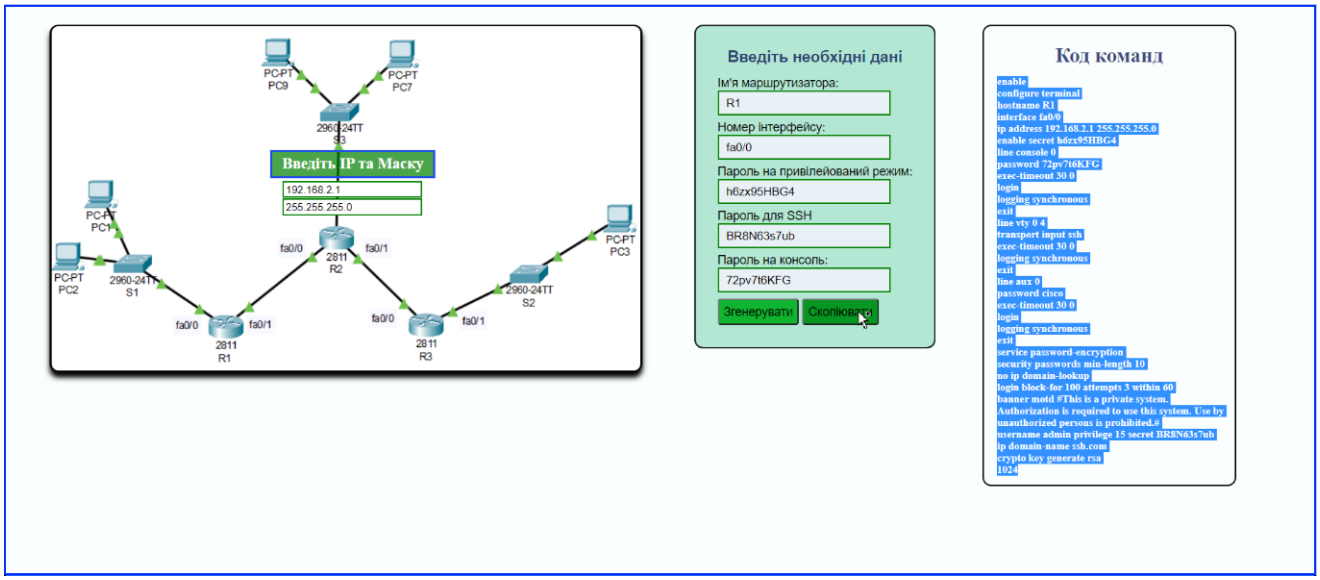

Рис. 3.6 – Дані для налаштування R1

Після того, як виконано генерацію команд на основі введених даних та натиснуто кнопку для створення частини сторінки з блоком, в якому розташовується код спостерігаємо такий текст, який в надалі потрібно використати для вводу в CLI(Рис. 3.7).

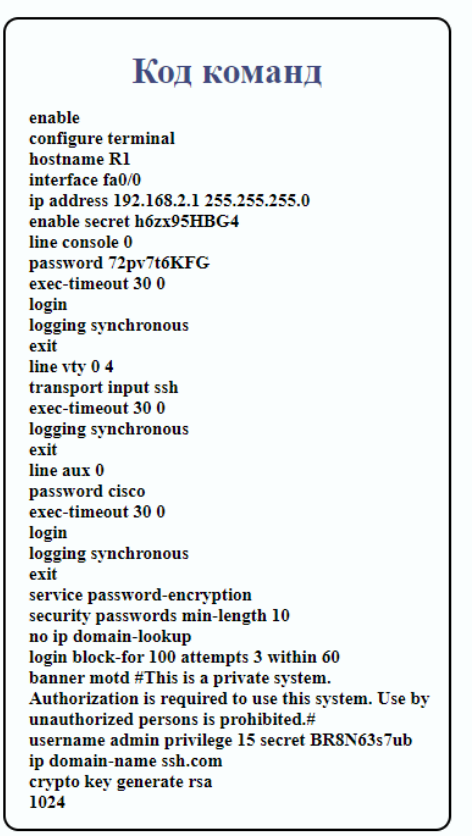

Рис. 3.7 – Команди для налаштування роутера R1

Після того, як команди було скопійовано в буфер обміну, потрібно перевірити коректність їх роботи в симуляторі Cisco Packet Tracer. Для цього відкриваємо саму програму та попередньо змодельовану мережу, знаходимо необхідний маршрутизатор, відкриваємо його налаштування(обрано роутер з ім’ям R1) та знаходимо пункт CLI в меню, після чого натискаємо на кнопку «Paste» в правому нижньому куті.

І на Рис. 3.8 спостерігаємо, що налаштування було успішно занесено в командний рядок та відсутні повідомлення про помилки.

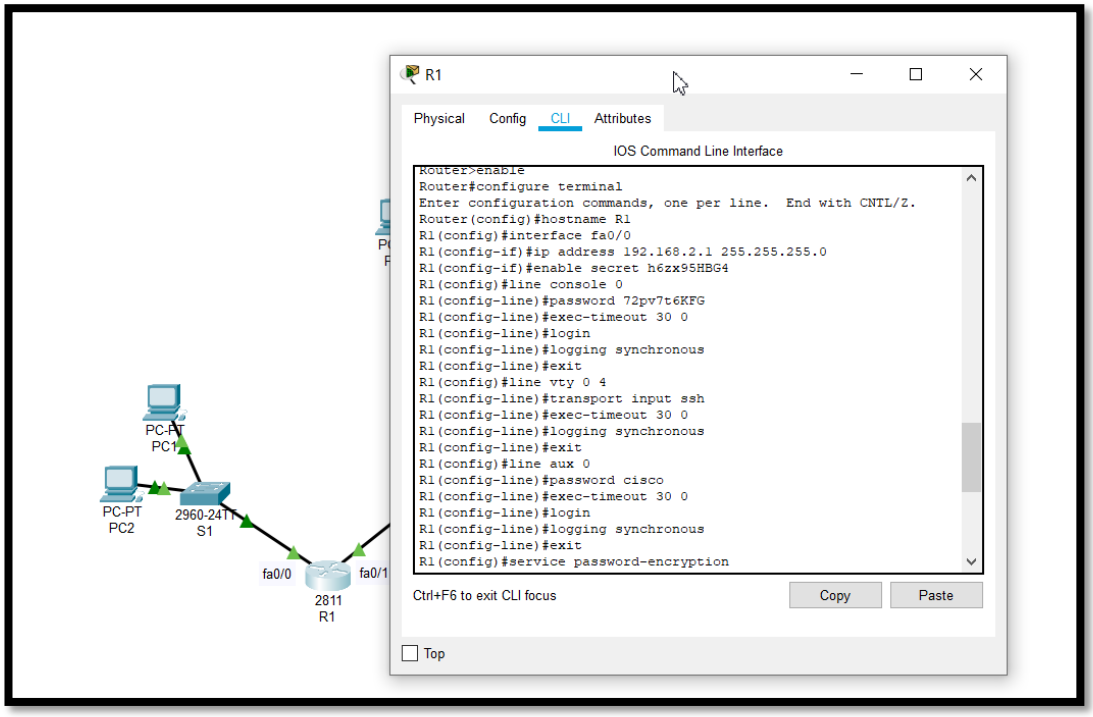

Рис. 3.8 – Використання скопійованих команд

Для перевірки того, що команди були застосовані на цьому маршрутизаторі потрібно в привілейованому режимі в CLI ввести команду «show running-config». Вона показує поточну конфігурацію приладу.

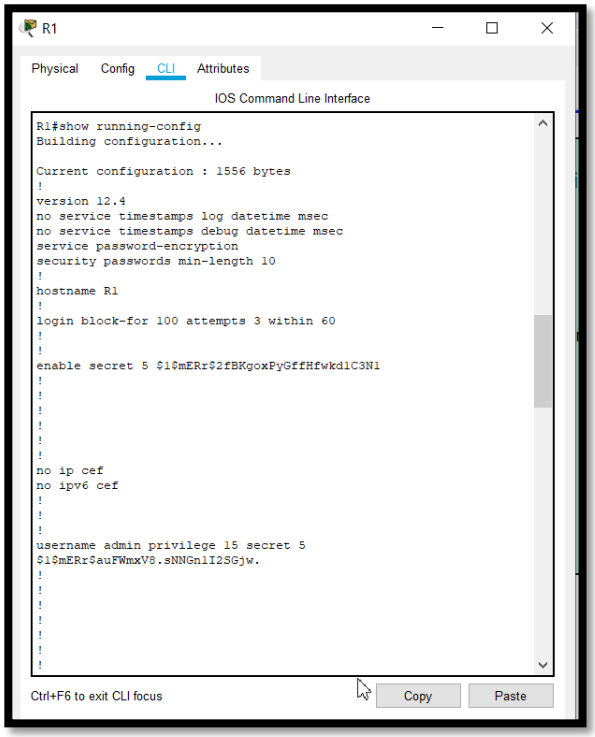

Рис. 3.9 – Виконання команди show running-config

Для того, щоб перевірити можливість підключення до маршрутизатора через протокол SSH необхідно в командному рядку комп'ютера ввести потрібну команду, а потім паролі для входу. Спочатку пароль від SSH, а потім для привілейованого режиму. При вході через комп'ютер також буде показаний банер, який попереджує нас про вхід в систему. На Рис. 3.10 показано порядок та результати підключення через віддалений доступ.

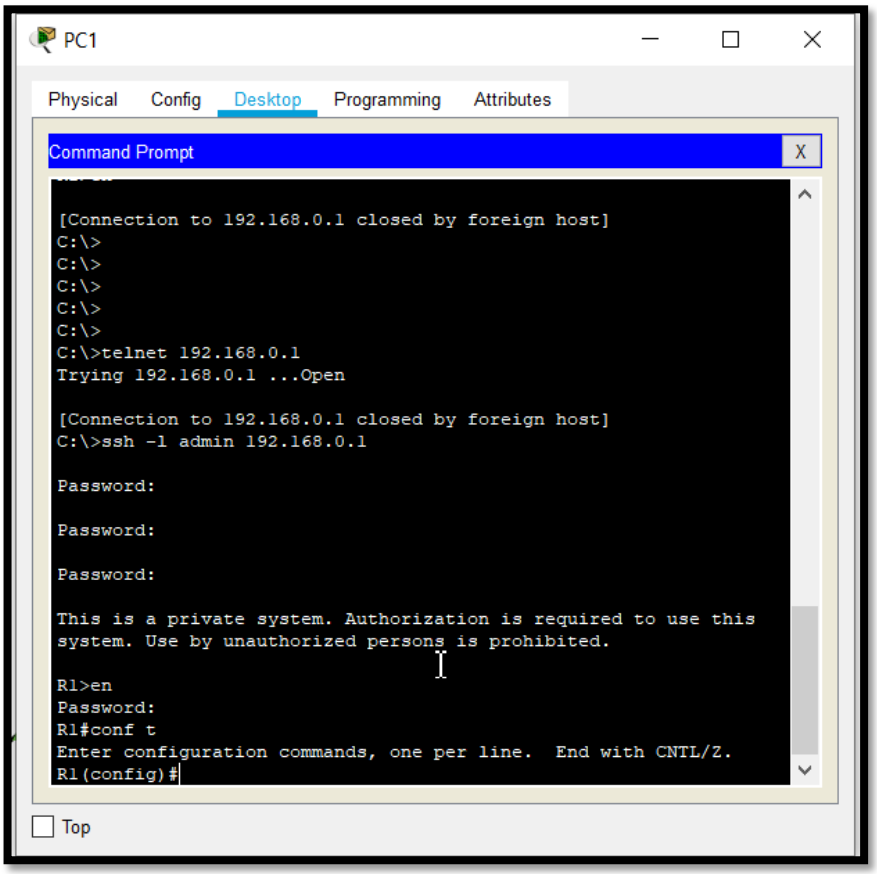

Рис. 3.10 – Перевірка підключення через протокол SSH

ВИСНОВКИ

У ході даної роботи було проведено аналіз інформації про методи захисту маршрутизаторів Cisco. Команди для підтримки безпеки зможуть надати можливість покращити стан системи та зменшити вплив зовнішніх факторів на неї. Важливо розуміти, що безпека мережі є дуже важливим фактором для її стабільної роботи. Тому потрібно намагатися використовувати більше різних способів для її усестороннього захисту.

У рамках роботи був розроблений графічний інтерфейс, який дозволяє налаштовувати маршрутизатори Cisco та генерувати код команд, які допомагають покращити захищеність мережі та зменшити час на пошук та введення необхідних даних. Була створена веб-сторінка з можливістю заповнення даних(паролі, адреси, імена), які потім підставляються у блок з деякою групою команд з можливістю копіювання та використання у власній мережі. За допомогою мови JavaScript було розроблено логіку роботи інтерфейсу, налагоджено валідацію введених даних, реалізовано генерацію коду та функцію збереження в буфер обміну необхідної інформації.

Розроблена система допомагає людям з початковими знаннями в даній області налаштовувати певні параметри безпеки маршрутизатора, а також надає можливість автоматизувати цей процес. Створений графічний інтерфейс може бути використаний, як в симуляторі, так і на реальному обладнанні мереж Cisco. Розроблена система показала гарні результати при перевірці роботи в програмі для моделювання Cisco Packet Tracer, що показує перспективу використання її на великій кількості маршрутизаторів для забезпечення їх налаштування в справжніх мережах.

Список літератури

1. Router Definition & Meaning [Електронний ресурс]: - Режим доступу: https://is.gd/HlXs6U
2. Cisco Systems, Inc [Електронний ресурс]: - Режим доступу: https://www.getwifi.ru/cisco.html
3. Cisco [Електронний ресурс]: - Режим доступу: https://www.cisco.com/
4. Барнс К. Защита от хакеров корпоративных сетей: ДМК Пресс - М., 2015.- 297 с.
5. Как обезопасить маршрутизатор Cisco [Електронний ресурс]: - Режим доступу: https://www.osp.ru/lan/2002/09/135242
6. Комп’ютерні мережі [Електронний ресурс]: - Режим доступу: https://web.posibnyky.vntu.edu.ua/fitki/3yarovijk_komp_merezhi/3.2.html
7. Cisco Packet Tracer - Базовые команды [Електронний ресурс]: - Режим доступу: https://open-networks.ru/d/16-cisco-packet-tracer
8. Service Password-Encryption Command on CISCO Router/Switch [Електронний ресурс]: - Режим доступу: https:// https://www.dmosk.ru/instruktions.php?object=cisco-ssh
9. Практическая безопасность сетей [Електронний ресурс]: - Режим доступу: http://blog.netskills.ru/2017/01/network-security11.html
10. Как настроить SSH на Cisco [Електронний ресурс]: - Режим доступу: https:// https://www.dmosk.ru/instruktions.php?object=cisco-ssh
11. SSH и OpenSSH: принципы работы, установка и настройка [Електронний ресурс]: - Режим доступу: http://bog.pp.ru/work/ssh.html
12. What is Cisco Packet Tracer [Електронний ресурс]: - Режим доступу: https://www.geeksforgeeks.org/what-is-cisco-packet-tracer/
13. What is JavaScript? [Електронний ресурс]: - Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

14. HTML Tutorial - W3Schools [Електронний ресурс]: - Режим доступу: https://www.w3schools.com/html/
15. Пфаффенбергер HTML, XHTML и CSS. Библия пользователя / Пфаффенбергер и др. - М.: Вильямс; Издание 3-е, 2015
16. Лазаро, Исси Коэн Полный справочник по HTML, CSS и JavaScript / Лазаро Исси Коэн, Джозеф Исси Коэн. - М.: ЭКОМ Паблишерз, 2018.- 631 с.
17. Clipboard.js — Copy to clipboard without Flash [Електронний ресурс]: - Режим доступу: https://clipboardjs.com/

ДОДАТОК А

```
new ClipboardJS('.copy_button');
hideComands();
//Сгенерувати команди
$('.generate_btn').click(showComandsIfNeed)
//Показати команди якщо все вірно
function showComandsIfNeed() {
    var ip = $('#ip').val();
    var mask = $('#mask').val();
    var isValidIp = validateIp(ip);
    var isValidMask = validateMask(mask);
    var isValidForm = validateForm();
    if (isValidIp == true && isValidMask == true && isValidForm == true) {
        showComands();
    } else {
        hideComands();
    }
}

//Функція показу блоку команд
function showComands() {
    $('.copy_button').css('cursor', 'copy');
    $('.copy_button').removeAttr('disabled');
    $('.result').show("slow");
    generate();
}
//Функція приховування блоку команд
function hideComands() {
    $('.copy_button').css('cursor', 'not-allowed');
    $('.copy_button').attr('disabled', 'disabled');
    $('.result').hide();
}
//Функція валідації ip адреси
function validateIp(ip) {
    var format_ip = /^([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])$/;
    if (ip.match(format_ip)) {
```

```
            $('#ip').removeClass("error");
            $('#ip').addClass("correct");
            return true;
        } else {
            $('#ip').removeClass("correct");
            $('#ip').addClass("error");
            return false;
        }
    }
    //Функція валідації маски підмережі
    function validateMask(mask) {
        var mask_array = ['255.255.255.255','255.255.255.254','255.255.255.252','255.255.255.248','255.255.255.240','255.255.255.224','255.255.255.192','255.255.255.128','255.255.255.0','255.255.254.0','255.255.252.0','255.255.248.0','255.255.240.0','255.255.224.0','255.255.192.0','255.255.128.0','255.255.0.0','255.254.0.0','255.252.0.0','255.248.0.0','255.240.0.0','255.224.0.0','255.192.0.0','255.128.0.0','255.0.0.0','254.0.0.0','252.0.0.0','248.0.0.0','240.0.0.0','224.0.0.0','192.0.0.0','128.0.0.0','0.0.0.0'];
        if (mask_array.includes(mask)) {
            $('#mask').removeClass("error");
            $('#mask').addClass("correct");
            return true;
        } else {
            $('#mask').removeClass("correct");
            $('#mask').addClass("error");
            return false;
        }
    }
    //Функція валідації форми
    function validateForm() {
        var name_router = $('#name_router').val();
        var num_int = $('#num_int').val();
        var priv_mod = $('#priv_mod').val();
        var ssh_mod = $('#ssh_mod').val();
        var console_mod = $('#console_mod').val();
        var name_pattern = /^[a-zA-z0-9]{1,}$/;
        var int_pattern = /^[a-zA-z]{1,3}[0-9]{1,3}\/[0-9]{1,3}$/;
        var password_pattern = /^[a-zA-z0-9@#$%^&*\.\-_]{10,}$/;
```

```
var valid = 0;
if (name_router.match(name_pattern)) {
    $('#name_router').removeClass("error");
    $('#name_router').addClass("correct");
} else {
    valid++;
    $('#name_router').removeClass("correct");
    $('#name_router').addClass("error");
}
if (num_int.match(int_pattern)) {
    $('#num_int').removeClass("error");
    $('#num_int').addClass("correct");
} else {
    valid++;
    $('#num_int').removeClass("correct");
    $('#num_int').addClass("error");
}
if (priv_mod.match(password_pattern)) {
    $('#priv_mod').removeClass("error");
    $('#priv_mod').addClass("correct");
} else {
    valid++;
    $('#priv_mod').removeClass("correct");
    $('#priv_mod').addClass("error");
}
if (ssh_mod.match(password_pattern)) {
    $('#ssh_mod').removeClass("error");
    $('#ssh_mod').addClass("correct");
} else {
    valid++;
    $('#ssh_mod').removeClass("correct");
    $('#ssh_mod').addClass("error");
}
if (console_mod.match(password_pattern)) {
    $('#console_mod').removeClass("error");
    $('#console_mod').addClass("correct");
} else {
```

```
        valid++;
        $('#console_mod').removeClass("correct");
        $('#console_mod').addClass("error");
    if (valid == 0) {
        return true;
    } else {
        return false;
    }
}
// Масив з даними для генерації
var code = [
"enable",
"configure terminal",
"hostname router_name",
"interface int_number",
"ip address ip_router mask_router",
"enable secret priv_mod",
"line console 0",
"password console_mod",
"exec-timeout 40 0",
"logging synchronous",
"login",
"exit",
"line vty 0 4",
"transport input ssh",
"exec-timeout 40 0",
"logging synchronous",
"exit",
"line aux 0",
"password cisco",
"exec-timeout 40 0",
"login",
"logging synchronous",
"exit",
"service password-encryption",
"security passwords min-length 10",
"no ip domain-lookup",
```

```
"login block-for 100 attempts 3 within 60",
"banner motd #This is a private system. This is a private system. You must be logged in to use this system.#",
"username admin privilege 15 secret ssh_mod",
"ip domain-name ssh.com",
"crypto key generate rsa",
"1024"
]
//Генерація команд
function generate() {
    var comands_output = "";
    var commands = [];
    commands = code;
    var ip = $('#ip').val();
    var mask = $('#mask').val();
    var router_name = $('#name_router').val();
    var int_number = $('#num_int').val();
    var priv_mod = $('#priv_mod').val();
    var ssh_mod = $('#ssh_mod').val();
    var console_mod = $('#console_mod').val();
    for (var i in commands) {
        var value = commands[i];
        value = value.replace('ip_router', ip);
        value = value.replace('mask_router', mask);
        value = value.replace('router_name', router_name);
        value = value.replace('int_number', int_number);
        value = value.replace('priv_mod', priv_mod);
        value = value.replace('ssh_mod', ssh_mod);
        value = value.replace('console_mod', console_mod);
        comands_output += '<h6>' + value + '</h6>';
    }
    document.getElementById('comands').innerHTML = comands_output;
}
```

ДОДАТОК Б

```
body{
    margin: 0;
    padding: 0;
}
input[type="text"]:focus{
    border: 2px solid #0031ff;
    outline:none;
}
.main_shema {
    display: block;
    margin-right: 20px;
    margin-left: 10px;
    border-radius: 10px;
    border: 2px dotted black;
    box-shadow: 0 6px 6px black;
    background: black;
}
.wrapper {
    padding-top: 20px;
    padding-bottom: 20px;
    border: 3px solid #0031ff;
    padding-bottom: 100px;
    background-color: rgba(241, 252, 252, 0.33);
}
.shema {
    display: inline-block;
    position: relative;
    float: left;
    margin-left: 40px;
}
#ip {
    position: absolute;
    top: 45%;
    left: 39%;
    font-size: 12px;
}
```

```
#ip.correct {
    border: 2px solid green;
}
#ip.error {
    border: 2px solid red;
}
#mask {
    position: absolute;
    top: 50%;
    left: 39%;
    font-size: 12px;
}
#mask.correct {
    border: 2px solid green;
}
#mask.error {
    border: 2px solid red;
}
.shema span {
    position: absolute;
    top: 36%;
    left: 37%;
    font-size: 18px;
    color: white;
    background-color: rgba(0, 128, 0, 0.71);
    padding: 3px 11px 3px 11px;
    font-weight: 700;
    border: 3px solid #1f4eea;
}
.form-style {
    display: inline-block;
    border: 2px solid rgba(0, 0, 0, 0.91);
    background: rgba(17, 180, 111, 0.3);
    border-radius: 10px;
    margin-left: 40px;
    margin-right: 40px;
    vertical-align:top;
```

```
}
.form-input {
    padding: 25px;
}
.form-input input[type="text"] {
    display: block;
    width: 80%;
    background: #e9f1f6;
    line-height: 25px;
    border-width: 1px;
    font-family: 'Roboto', sans-serif;
    padding: 0 8px;
    margin-bottom: 4px;
.form-input input[type="text"].error {
    border: 2px solid red;
}
.form-input input[type="text"].correct {
    border: 2px solid green;
}
.form-input input[type="button"] {
    display: inline-block;
    background: #11b430;
    border-radius: 3px;
    font-size: 14px;
    outline: none;
    padding: 5px;
    margin-top: 3px;
}
.form-input input[type="button"]:hover {
    background: rgb(8, 150, 35);
    cursor: pointer;
}
.copy_button.block{
cursor: not-allowed;
    background: black;
}
.form-input h3 {
```

```
    margin-top: 0px;
    margin-bottom: 10px;
    text-align: center;
    font-family: 'Roboto', sans-serif;
    font-weight: 300;
    font-size: 18px;
    color: rgba(2, 11, 82, 0.75);
    font-weight: bold;
}

.form-input label {
    font-size: 14px;
    font-family: 'Roboto', sans-serif;
}
.result {
    display: inline-block;
    border: 2px solid black;
    padding: 10px;
    margin-left: 10px;
    margin-right: 50px;
    border-radius: 10px;
    width: 270px;
.result h2 {
    padding: 10px;
    text-align: center;
    margin: 0px;
    color: rgba(2, 11, 82, 0.75);
}
h6 {
    padding-left: 5px;
    margin: 0;
    font-size: 12px;
}
#footer {
    display: block;
    background-color: rgba(0, 69, 255, 0.78);
    padding: 20px;
```

```
    font-family: 'Roboto', sans-serif;
    font-size: 20px;
    font-weight: bold;
    border: 3px solid #000000;
    border-top: none;
    text-align: center;
}
```

ДОДАТОК В

```
<!DOCTYPE html>
<html>
<head>
   <link href="css/style.css" rel="stylesheet" type="text/css" />
   <script type="text/javascript" src="js/jquery-3.5.1.min.js"></script>
   <meta charset="utf-8">
   <title>Графічний інтерфейс</title>
</head>
<body>
   <div class="wrapper">
      <div class="shema">
         <img class="main_shema" src="photo/main_schema.png">
         <span>Введіть IP та Маску</span>
         <input type="text" id="ip" placeholder="IP: xxx.xxx.xxx.xxx">
         <input type="text" id="mask" placeholder="MASK: xxx.xxx.xxx.xxx">
      </div>

      <form class="form-style" method="post">
         <div class="form-input">
            <h3>Введіть необхідні дані</h3>
            <label for="name_router">Ім'я маршрутизатора:</label>
            <input type="text" placeholder="Приклад: R1" id="name_router">
            <label for="num_int">Номер інтерфейсу:</label>
            <input type="text" placeholder="Приклад: fa0/0" id="num_int">
            <label for="priv_mod">Пароль на привілейований режим:</label>
            <input type="text" placeholder="Пароль(>9 символів)" id="priv_mod">
            <label for="ssh_mod">Пароль для SSH </label>
            <input type="text" placeholder="Пароль(>9 символів)" id="ssh_mod">
            <label for="console_mod">Пароль на консоль: </label>
            <input type="text" placeholder="Пароль(>9 символів)" id="console_mod">
            <input type="button" class="generate_btn" value="Згенерувати">
            <input type="button" value="Скопіювати" class="copy_button" data-clipboard-
target=".comands">
         </div>
      </form>
      <div class="result" id="result">
```

```
                <h2>Код команд</h2>
                <div class="comands" id="comands">
                </div>
            </div>
        </div>
        <div id="footer">
            <p>Графічний інтерфейс для налаштування параметрів безпеки та аутентифікації користувача на маршрутизаторах Cisco</p>
        </div>
        <script type="text/javascript" src="js/clipboard.js"></script>
        <script type="text/javascript" src="js/script.js"></script>
    </body>
</html>
```