МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Графічний інтерфейс для налаштування безпеки портів комутаторів»

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

Керівник роботи — **Великодний Д.В.**

Студентки групи ІН.м – 81н — **Оводова Н.Б.**

СУМИ 2020

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«Інформатика»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Оводовій Наталії Борисівні
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Графічний інтерфейс для налаштування безпеки портів комутаторів*
затверджую наказом по інституту від "____" _____ 20 ___ 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 проблеми. Постановка задачі дослідження. 2) Визначення застосування функції безпеки портів комутаторів 3) Відтворення комп'ютерної мережі з використанням безпеки портів 4) Розроблення веб-додатку з інформаційно-графічним інтерфейсом для комутаторів з налаштуванням функції безпеки портів.*

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 проблеми. Постановка задачі дослідження*		
2.	*Визначення застосування функції безпеки портів комутаторів.*		
3.	*Відтворення комп'ютерної мережі з використанням безпеки портів*		
4.	*Розроблення веб-додатку з інформаційно-графічним інтерфейсом для комутаторів з налаштуванням функції безпеки портів.*		
5.	*Оформлення пояснювальної записки до дипломної роботи*		

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис)

РЕФЕРАТ

Записка: 63 стор., 34 рис., 2 табл., 1 додаток, 16 джерел.

Мета роботи — розроблення графічного інтерфесу, який дозволяє налаштування конфігурації інтерфейсів комутаторів CISCO з підтримкою функції безпеки портів.

Об'єкт дослідження — безпека мережі комп'ютерні на основі комутаторів CISCO.

Предмет дослідження — функція безпеки портів комутаторів.

Методи дослідження — відтворення в симуляторі Cisco Packet Tracer мережі з функцією безпеки портів комутатора.

Результати —графічний інтерфейс, що був запрограмований на основі веб-орієнтованої інформаційної системи, яка дозволяє автоматичне конфігурування інтерфейсів комутатора з підтримкою функції безпеки портів комутатора. Додаток має підтримувати генерацію налаштувань та копіювання їх на реальне обладнання комутаторів. Додаток було розроблено як веб-додаток за допомогою фреймворку Vue Js.

PORT-SECURITY, ACL, DHCP SNOOPING, КОМУТАТОРИ, ВЕБ-ОРІЄНТОВАНА СИСТЕМА, ВЕБ-ДОДАТОК, ГРАФІЧНИЙ ІНТЕРФЕЙС, СИМУЛЯТОР, VUE JS, CISCO PACKET TRACER.

ЗМІСТ

ВСТУП

Комп’ютерна мережа є найголовнішим засобом розповсюдження та зберігання інформації для корпорацій та підприємств, яка може в собі зберігати інформацію про своїх клієнтів, про працівників та про різні розробки чи продукти. Тому виникає потреба в забезпеченні та налаштуванні мережі аби данні не могли перехопити зловмисники. Досить багато було зроблено досліджень та впроваджено програмних продуктів для налаштування безпеки комутаторів. Відомі програмісти зробили вже влаштовану функцію для забезпечення безпеки портів, але їм не вдалось зробити веб-додаток, який би забезпечував конфігураційні команди для мануального налаштування в залежності від вимог підприємства чи компанії.

У роботі досліджено функцію безпеки портів та режими реагування комутаторів на небезпеку, що виникає в мережі. Режими мають бути досліджені зі сторони телекомунікацій на підприємстві.

Важливо зазначити, що в залежності від бажаного результату при несанкціонованих атаках на комутатори ми маємо використовувати різні режими для налаштування комутаторів, це і є протиріччям, що виникає в даній роботі.

Якщо вірно використовувати режими реагування на небезпеку для портів і при цьому враховувати особливості комутаторів, то можна спрогнозувати що комутатори будуть відповідати всім поставленим вимогам.

Допрацьований метод налаштування безпеки портів комутаторів, де всі налаштування мають виконуватись мануально, а отже я реалізую веб-додаток, що допоможе системному адміністратору облегшити налаштування безпеки портів лише за натисненням кнопки для генерації відповідних налаштувань.

Завдяки веб–додатку, налаштування безпеки портів комутаторів буде ще швидшим. Це вирішить проблему, що постала в роботі. Буде проведений детальний розбір методів для забезпечення безпеки комутаторів, та буде обраний найбільш відповідний поставленій задачі.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ НАЛАШТУВАННЯ БЕЗПЕКИ ПОРТІВ КОМУТАТОРІВ. ОСНОВНІ АЛГОРИТМИ.

На цей час відомо про досить незначну кількість методів для налаштування безпеки портів комутаторів. Однак можна виділити 3 найбільш популярних та відомих способів, які допоможуть захистити з`єднання комутатору та комп`ютерів від несанкціонованого доступу (рис. 1. 1).

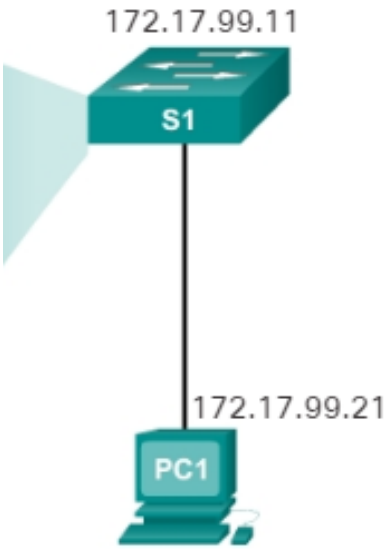

Рисунок 1.1 – Приклад з`єднання комутатора та комп`ютера [1]

Access Control List – це метод безпеки, що дозволяє обрати трафік за деякими критеріями, а те що не сходиться з цими критеріями буде відкидатись. ACL необхідно буде фільтрувати трафік і той що приходить до нас, і той що йде від нас. Маршрутизатор має досконало розпізнати кожен отриманий пакет, та прийняти вірне рішення чи потрібно пересилати його далі чи знищити. Прийняття рішення відбувається на основі фільтрування трафіку за поданими критеріями:

- Адреса звідки йде трафік
- Верхнього рівня протокол
- Кінцева точка трафіку

Щоб активувати ACL його потрібно мануально створити в будь-якому редакторі, а потім додати до налаштувань інтерфейсу [2].

DHCP Snooping – є високоефективною технологією, що потрібна для захищення комутатора від підміни його DHCP (Dinamic Host Configuration Protocol) на несправжній, а саме той, що має використовуватись для проникнення до вашої мережі. Ця технологія не дозволить зловмиснику підмінити IP адреси вашого комп`ютера. Суть технології полягає у тому, щоб обмежити на заданих портах опрацювання запитів, що прийшли від DHCP та відкинути їх. Також є можливість створювати власну базу даних з ненадійними хостами та IP адресами [3].

Port Security – це функція, яка налаштовується на комутаторі за допомогою декількох команд для заборони доступу до комутатора усім MAC – адресам, які відсутні в конфігураційній таблиці і не були сконфігуровані на інтерфейсі комутатора динамічно чи статично. Отже, функція створена для захисту інтерфейсу від забороненої зміни адреси через підключення до інтерфейсу. Тобто функція виконує головну роль в роботі комутатора та його безпеки, адже попереджає можливість несанкціонованих проникнень для викрадення та знищення інформації, що зберігається на пристрої, що був сконфігурований в таблиці комутатора.

Також ця функція є корисною, якщо потрібно ввести точну кількість доступних MAC-адрес для кожного інтерфейсу окрему, але у разі необхідності це число можна збільшити або зменшити. [4].

Це досить нескладний метод, що використовується для в багатьох організаціях. Тому потрібно буде лише відключити всі порті комутаторі що не будуть використовуватись, а ті що будуть залучені потрібно буде налаштувати певним чином відповідно до заданих вимог для системного адміністратора. Наприклад, якщо взяти комутатор Cisco 2950-24, то в ньому буде 24 порти для Fast Ethernet, та є ще 2 GigabitEthernet, які нам не потрібно використовувати. Отже, якщо буде залучено лише 4 порти, то 20 портів, що залишились повинні бути відключені, бо не будуть використовуватись. На рис. 1.2 показано активування

порту за допомогою командою «no shutdown» і в результаті порт буде активований.

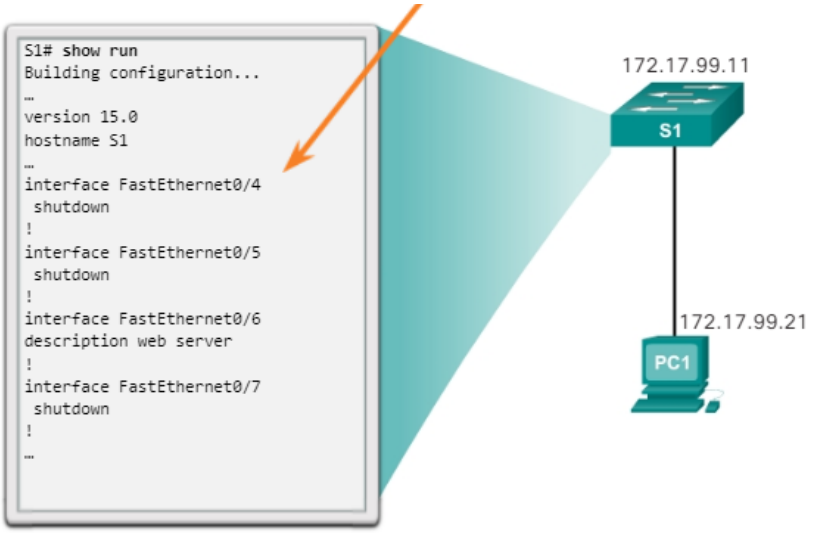

Рисунок 1.2 – Активування порту командою «no shutdown»[1]

Подану команду можна використовувати як для кожного порту окремо, так можна і вказувати діапазон для портів комутатору.

1.1 ACL для налаштування безпеки портів

На сьогодні відомо про ACL лише стандартні і розширені. Якщо перед нами постає завдання для фільтрації трафіку лише за IP адресою, того хто відправляє трафік, то це буде стандартний вид, якого не завжди буває достатньо. Бо це не буде гарантувати безпечний фільтр для мережі, тому їх зазвичай використовують для всієї мережі, тобто або все дозволено або заборонено.

Отже, потрібно шукати розв`язання цієї проблеми, щоб можна було фільтрувати за значно більшою кількістю ознак, і це буде ефективніше та безпечніше для мережі в цілому. Зазвичай, процес налаштування стандартної ACL є досить нескладним навіть для недосвідченого користувача.

Приклад, налаштування мережі зі стандартним ACL (рис. 1. 3).

На рисунку міститься команда яка забезпечує доступ до вказаних безпечних хостів, які можна вводити мануально, а інші будуть заборонені.

```
access-list 1 permit host 192.168.10.50
access-list 1 permit host 192.168.10.53
access-list 1 permit host 192.168.10.60
```

Рисунок 1.3 – Стандартний ACL [2]

ACL, що зображений на рисунку 1.3 лише для декількох ip адрес дозволяє вихід до мережі Інтернет [2].

Варто, зазначити що також може використовуватись розширений ACL, який може фільтрувати трафік за значно більшою кількістю параметрів, які є важливими для безпечного трафіку:

- Номери інтерфейсів;
- Тип трафіку для протоколу;
- Порт одержувача;
- Протокол;
- Адреса того отримав;
- Порт відправника;
- Адреса того хто відправив.

Тобто, розширений ACL має досить велику кількість критеріїв, що також можна за потреби розширювати, це і робить цей тип ACL більш використовуваним. Його можливо розширити за допомогою інших телекомунікаційних технологій, таких як:

- Time Based – ACL, яка буде корисною в офісах та навчальних класах університетів, бо фільтрування трафіку в ній працюють лише у заданий час. Тому варто застосовувати цю ACL для відключення Інтернету в неробочий час та злагодженої роботи в робочий час.

- Dynamic – ACL, де за замовчуванням конфігураційні налаштування спочатку будуть вимкнені та не доступні, але коли відбувається підключення до маршрутизатора з правами адміністратора, то рядки автоматично активуються. Це було зроблено для того, аби не перевантажувати локальну мережу налаштуваннями без потреби, а лише тоді коли відбудеться вихід в мережу Інтернет.

- Reflexive – ACL який називають дзеркальним, бо має можливість зберігати порти та IP адреси, які контактували з іншими адресами з інших мереж. Та на основі цього повинен формуватись цей список доступу, що дозволяє

зворотню відповідь від іншої мережі, але лише тоді, якщо до неї вже звертались з нашої мережі [5].

Отже, використання списку доступів є ефективним методом для забезпечення безпеки портів лише за допомогою кількох критеріїв. Стандартний вид є легшим, бо є можливість редагування, видалення рядків, якщо вони мали свої унікальні номери, та в додаванні ще рядків.

Для забезпечення кращої безпеки рекомендовано використовувати дзеркальні ACL. Через те що цей вид є безпечним, та здійснює контактування лише з відомими раніше мережами, а не лише фільтрує за критеріями. Нижче поданий приклад дзеркальної ACL (рис. 1. 4).

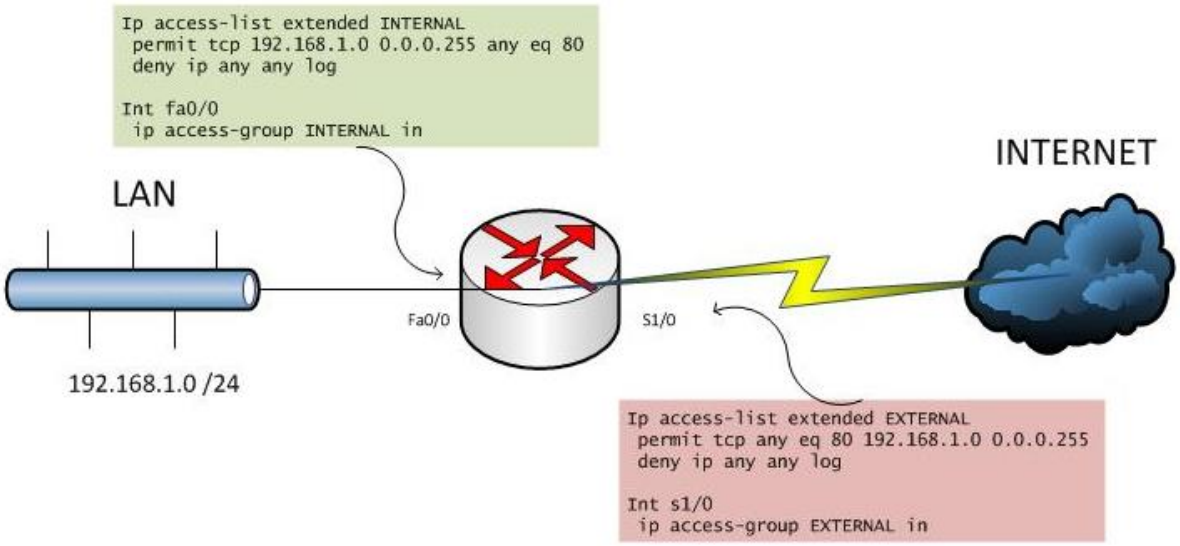

Рисунок 1.4 – Дзеркальні ACL [5]

На зеленому фоні можна побачити «internal ACL», що означає що це внутрішня конфігурація ACL, яка має пройти успішно перевірку і якщо це відбулось то ці налаштування (але вже для іншого порту) мають з’явитись для «external ACL» та повинна відбутись тепер перевірка вже з цим ACL.

Все ж таки метод є досить складним для розуміння як початківцю так і досвідченому фахівцю, адже вимагає досконалого розуміння мережі та її слабкі місця, для подальшого використання цих списків доступу, що можуть фільтрувати трафік лише за поданими критеріями. Тому при невірному визначенні критеріїв мережа стає уразливою до атак [6].

1.2 Функція безпеки портів Port-Security.

Усі порти комутаторів (їх ще називають інтерфейси) повинні бути захищені перед тим, як комутатор буде розгорнутий для використання у виробництві. Один із способів захисту портів – це реалізація функції, яка називається безпекою портів. Для цього слід вказати точно скільки MAC адрес буде дозволено порту. Таким чином інтерфейс буде приймати пакети від дозволених адрес, які були сконфігуровані, а пакети від адрес що не входять до цього списку будуть відкидатись та не будуть прийматись комутатором.

Безпека одного порту може бути налаштована таким чином, щоб дозволити приймати трафік від одного до кількох статичних або динамічних MAC адрес.Тобто якщо встановлена кількість MAC адрес та задані явно ці адреси, з якими повинен працювати порт, то тільки пристрої з цими конкретними MAC адресами можуть успішно увімкнутися до порту. Варто зазначити, що це ніяк не впливає на швидкість та злагодженість роботи як комутатора так і порту [1].

Для спричинення ситуації, коли спрацьовує режим реагування на небезпеку потрібно щоб всі порти комутатора мали налаштування безпеки портів, та мали заповнену таблицю конфігурації, де всі дозволені інтерфейсам адреси були або автоматично заповнені або вручну системним адміністратором. І коли кількість адрес стає більшою або не сходиться з дозволеними, то порушується режим безпеки та вони не будуть мати можливість з’єднання до мережі комутатора (рис. 1. 5).

Рисунок 1.5 – Налаштування безпеки портів с прив`язкою до MAC адрес [1]

Конфігурація безпеки порту має можливість підтримувати такі типи MAC адрес:

- Sticky (захищена) MAC адреси – адреси, які можуть бути і статичні і динамічні. Їх головна особливість, що вони визначаються вже в самому процесі роботи мережі. Цей тип може бути корисним коли в самому початку адреси є невідомими, а стають відомими лише в процесі
- Static (статична) MAC адреси – це адреси, які мануально конфігуруються на порту системним адміністратором, та зберігаються в конфігурації комутатора, як дозволені адреси. Такий тип адрес доцільно використовувати власникам великих підприємств, де адреси є зарезервованими, та їх досить легко занести до налаштувань.
- Dynamic (динамічна) MAC адреси – адреси, які точно не можна вказати чи задати, бо вони стають відомими лише після запуску комутатора та його налаштування, і лише потім будуть додані до комутатора. Такі адреси можливо використовувати на тих підприємствах, де неважливо які саме адреси будуть мати доступ а важлива лише їх кількість, тому такі адреси будуть вже не доступні після перезапуску інтерфейсів чи комутатора в цілому [7].

Захищені MAC адреси зазвичай налаштовуються автоматично та мануально. Для автоматичного налаштування необхідно ввімкнути на інтерфейсі sticky learning (навчання для захищених MAC адрес). Та за допомогою спеціальних команд динамічні адреси будуть запам’ятовуватись на комутаторі. Навчання повинно вмикатися на кожному порту окремо, та за допомогою певного набору конфігурацій [3].

«switchport port-security mac-address sticky mac-address» команда вводиться в консоль комутатора для налаштування захищеної MAC-адреси вручну. Для динамічного налаштування команда та ж сама, але не потрібно вказувати MAC-адресу.

Коли використовується захищена адреса, то в разі виникнення непередбачуваних ситуацій, таких як технічні помилки, переривання з’єднання та при перезапуску інтерфейсу комутатора, ці адреси все одно будуть збережені у конфігураційних налаштуваннях. Це означає що інтерфейс може не виконувати

повторне визначення адрес після вмикання інтерфейсу, а вже їх має в таблиці конфігурації. Але, у випадку, коли адреси все ж таки були втрачені з якихось причин, то їх не можливо буде відновити [8].

Режимів реагування на небезпеку відомо також три:

- «switchport port-security violation restrict» в цьому режимі реагування також будуть відкинути пакети від невідомих адрес, але також не будуть генеруватись повідомленняпро помилку, які сигналізують про небезпеку, але при цьому збільшується лічильник для підрахунку порушень при кожному виявленні несанкціонованого доступу.
- «switchport port-security violation shutdown» при тому що були виявлені несанкціонованих підключень інтерфейс будуть вжиті негайні заходи по захисту мережі, які проявляються в відключенні порту на якому було виявлено підключення. В цьому випадку не генерується повідомлення про помилку, але збільшується лічильник для підрахунку порушень. Для подальшої роботи потрібно знову ввімкнути інтерфейс за допомогою команд «shutdown» .
- «switchport port-security violation protect» при тому що були виявлені невідомого підключення до інтерфейсу, буде здійснюватись захист комутатора таким чином, що будуть ігноруватись та знищуватись всі пакети, що будуть надходити з невідомої адреси, при цьому не відправляються повідомлення про помилку, та не збільшується лічильник для підрахунку порушень і не відключається порт [9].

Якщо буде спровокована несанкціонована атака на комутатор, то при вірному налаштуванні функції port-security буде з’являтись повідомлення про небезпеку (рис. 1.6).

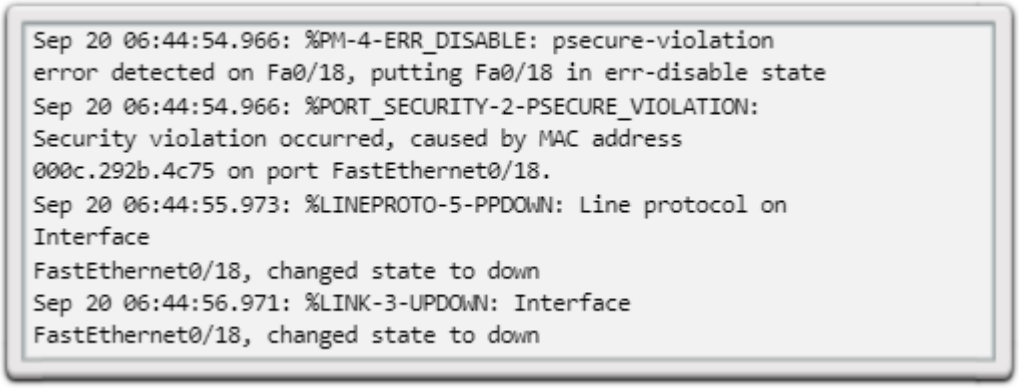

Рисунок 1.6 – Повідомлення про несанкціоновану атаку [1]

В повідомленні міститься дата, коли була спровокована небезпека, повідомлення та сам порт, на якому відбувалась атака на комутатор. Як видно, то це досить зрозуміле повідомлення, яке було видано системою.

В таблиці 1.1 показано для більшого розуміння принцип роботи режимів реагування.

Таблиця 1.1 Режими порушення безпеки [10]

Режим реагування	Пересилання трафіку	Підтримка повідомлень SYSLOG	Повідомлення про помилку	Збільшення лічильника порушень	Виключення порту
Protect	Ні	Ні	Ні	Ні	Ні
Restrict	Ні	Так	Ні	Так	Ні
Shutdown	Ні	Ні	Ні	Так	Так

Який саме режим реагування обрати – це вже рішення яке має приймати самостійно системний адміністратор, опираючись на завдання, що постали для організації. Та не зважаючи на те, який тип режиму реагування на небезпеку буде обраний, потрібний рівень безпеки буде належним чином наданий.

Підводячи висновки, для реагування на небезпеку є такі ситуації:

- Випадок, коли невідомий для комутатора MAC–адрес приймає спроби для заволодіння доступу до інтерфейсу, де було вже налаштовано задана кількість динамічних MAC-адрес;

- Випадок, коли статична адреса була закріплена за певним пристроєм та міститься в таблиці конфігурації комутатора. І ця ж адреса намагається бути використана на іншому комп'ютері, тим самим порушуючи режим реагування на небезпеку [10].

1.3 DHCP Snooping.

Налаштована конфігурація на комутаторі для підтримки DHCP Snooping є потужним інструментом для захисту мережі від атак, що призводять до втрати інформації. Принцип роботи технології є досить простим та полягає в класифікуванні інтерфейсів комутатора на 2 категорії: надійні та ненадійні порти. Надійний порт містить в собі ті повідомлення, що можна вважати за ті, які вважаються надійними. Ненадійний порт містить повідомлення, що не можна вважати за надійні [3].

Процес впізнання надійних та ненадійних портів полягає у створенні таблиці DHCP (рис.1.7). У цю таблицю буде записано MAC-адресу, IP адресу, час аренди в секундах, тип, VLAN, та помітки для інтерфейсу. У випадку, якщо отриманий пакет буде не збігатись з тим, що знаходиться в таблиці, то він буде видалений.

	MAC Адрес	IP Адрес	Аренда(сек)	Тип	VLAN	Интерфейс
Entry 1	e4-54-e8-9d-ab-42	10.32.96.19	2673	dhcp-snooping	10	Eth 1/23
Entry 2						
Entry 3						
...						

Рисунок 1.7 – Заповнення таблиці для DHCP Snooping [3]

Серед основних атак, з якими може впоратися ця технологія є:

- DHCP Starvation – підробна система буде відсилати на DHCP сервер нескінченну кількість запитів на отримання IP адреси за допомогою сфабрикованих MAC-адрес, з метою переповнення серверу DHCP запитами, для того аби закінчився пул можливих адрес

- Спуфінгова атака – якщо шахрай зробить свій шлюз, як шлюз за замовчуванням та намагається зробити підроблену відповідь на DHCP запити, і тим самим реалізовує аткаку через посередника. Можливий перехват трафіку від користувачів [11].

Налаштування технології є простим і полягає у конфігуруванні комутатора за допомогою декількох команд, та вказування мережі для якої ми виконали налаштування.

1.4 Постановка задачі

Зараз майже всі великі корпорації та компанії мають велику кількість офісів та філіалів, які з'єднані між собою локальною мережею та мережею Інтернет. Тому більшість працівників має доступ до мережі Інтернет і може використовувати її як для ефективної роботи так і як засіб для дозвілля, та у особистих цілях. Ваша мережа уразлива та до неї може підключитися особа, що не має жодного відношення до компанії, і є зловмисником, що хоче перехопити трафік та дізнатися секретну інформацію як компанії, так і особисту інформацію працівників. Тому на кожному підприємстві постає задача в забезпеченні безпеки цілісності даних що передаються в мережі.

Отже, після детального аналізу методів, що описані в роботі, було вирішено, що найліпшим рішенням буде застосування налаштування функції безпеки портів. За допомогою цієї функції потрібно зробити веб–додаток що містить графічний інтерфейс, який має генерувати конфігураційні команди для автоматичного налаштовування інтерфейсів комутаторів Cisco.

Розроблений додаток повинен вирішувати проблему налаштування комутатора, та повинен дозволяти і впевненим фахівцям і новачкам налаштовувати комутатори на реальному обладнанні компанії Cisco та на симуляторі.

Інтерфейс повинен бути розроблений зрозумілим і недосвідченому користувачу, що тільки починає знайомитись з телекомунікаційними

технологіями а також може не мати потрібного для роботи досвіду у використанні таких веб-інтерфейсів.

Результатом розроблення графічного інтерфейсу є реалізована веб–сторінка, що містить в собі форму для занесення відповідних вхідних даних, та отримання налаштувань для комутатора, які також можуть бути перенесені до буфера обміну завдяки копіюванню та вставити в справжній комутатор чи в Cisco Packet Tracer для симульованого комутатора.

Постановка задачі:

1. Налаштування безпеки портів комутатора в симуляторі Cisco Packet Tracer.
2. Розроблення графічного інтерфейсу з налаштуванням безпеки портів .
3. Тестування веб–додатку в симуляторі Cisco Packet Tracer.

2 МОДЕЛЮВАННЯ ФУНКЦІЇ БЕЗПЕКИ ПОРТІВ З ВИКОРИСТАННЯМ СИМУЛЯТОРА CISCO PACKET TRACER ТА КОМУТАТОРІВ CISCO

2. 1 Конфігурація безпеки портів на базі комутаторів Cisco

Cisco Packet Tracer – це симулятор, який використовується для побудування мереж різних ступенів складності, ієрархії та методів. Використовується як студентами вузів, які хочуть закріпити свої знання з телекомунікаційних наук, а також і досвідченими спеціалістами для моделювання мережі в організації та за її межами. Підтримує майже всі складні сценарії та методи. Допомагає викладачам з усього світу в полегшенні викладанню матеріалу і вивченні складних сценаріїв, які розвивають розуміння складу мережі та принципу її роботи.

Packet Tracer може повноцінно замінити навчання на реальному обладнанні, через те що має зрозумілий навіть новачку інтерфейс, який максимально відповідає реальному обладнанню, та його комплектації [12].

Перевагою цього симулятора є те, що він зазвичай безкоштовно має надаватись студентам, викладачам, системним адміністраторам, та просто людині, яка хоче відкрити для себе світ телекомунікаційних технологій.

Симулятор Cisco Packet Tracer має нескладний для розуміння інтерфейс, що містить в собі всі необхідні в роботі інструменти. Усі вони знаходяться у лівому нижньому кутку, а поруч також можна знайти усі інші пристрої відповідно до категорії:

1. З’єднання
2. Телевізор
3. Сніффер
4. Хаби
5. Комп’ютери та ноутбуки
6. IP-телефон
7. Маршрутизатори
8. Комутатор.

9. Сервери

Тому було обрано саме Cisco Packet Tracer, бо завдяки йому добре видно основні принципи налаштування безпеки портів, які описані в табл.1.1.

Таблиця 1. 1 Основні команди для налаштування інтерфейсу Cisco [4].

Етапи	**Команда**
Вкажіть інтерфейс для котрого необхідно налаштувати безпеку порту.	S1(config)#int range fa 0/1-2
Налаштуйте режим інтерфейсу в режимі доступу.	S1(config–if)#switchport mode access
Включіть безпеку портів на інтерфейсі.	S1(config–if)#switchport port–security
Задайте максимальну кількість MAC адрес.	S1(config–if)#switchport port–security maximum X
Задайте режим реагування на небезпеку.	S1(config–if)#switchport port–security violation re-strict/shutdown/protect
Задайте sticky MAC адресу.	S1(config–if)#switchport port–security mac–address sticky.
Задайте MAC-адресу.	S1(config–if)#switchport port–security mac–address XXXX.XXXX.XXXX

Отже, розглянемо налаштування комутатора таким, чином коли до комутатора буде увімкнено невідомий йому MAC адрес та за допомогою різних режимів реагування буде впроваджена поведінка для комутатора в мережі. Було обрано модель комутатора Cisco 2950-24, в ньому є 24 порти типу Fast Ethernet та 2 типу Gigabit Ethernet, але для роботи ми оберемо лише 2, адже функція безпеки портів буде дублюватись для інших інтерфейсів комутатора. Також буде оглянуто всі режими реагування на небезпеку, щоб мати повне розуміння роботи системи в цілому. Комп’ютери були обрані звичайні PC 1 та PC 2, та ще один ноутбук, який

має симулятор. На них були задані ip адреси, але MAC- адреса було вказана лише для одного з них (рис. 2.1).

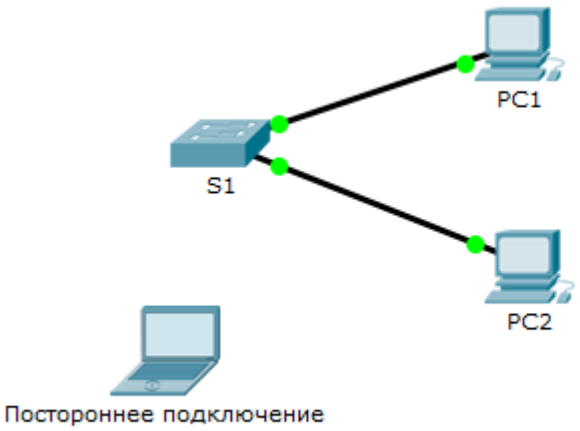

Рисунок 2.1 – Підключення двох комп'ютерів та одного несанкціонованого підключення до інтерфейсу комутатора

Конфігурація комутатора з режимом Restrict:

en

conf t

int fa 0/1 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation restrict – задаємо режим доступу Restrict.

switchport port-security mac-address 0005.5E80.22A3

exit

int range fa 0/3-24- вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/1 – перевірка стану порту.

Аналогічні дії потрібно виконати і для fa 0/2:

en

conf t

int fa 0/2 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation restrict – задаємо режим доступу Restrict.

switchport port-security mac-address sticky

exit

int range fa 0/3-24 – вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/2 – перевірка стану порту.

Поглянемо на результат за допомогою команди sh run (рис. 2.2). Ця команда дає змогу подивитись налаштування, які були сконфігуровані на комутаторі в режимі реального часу. Для розгорнення всіх портів потрібно натискати кожного разу на пробіл на клавіатурі.

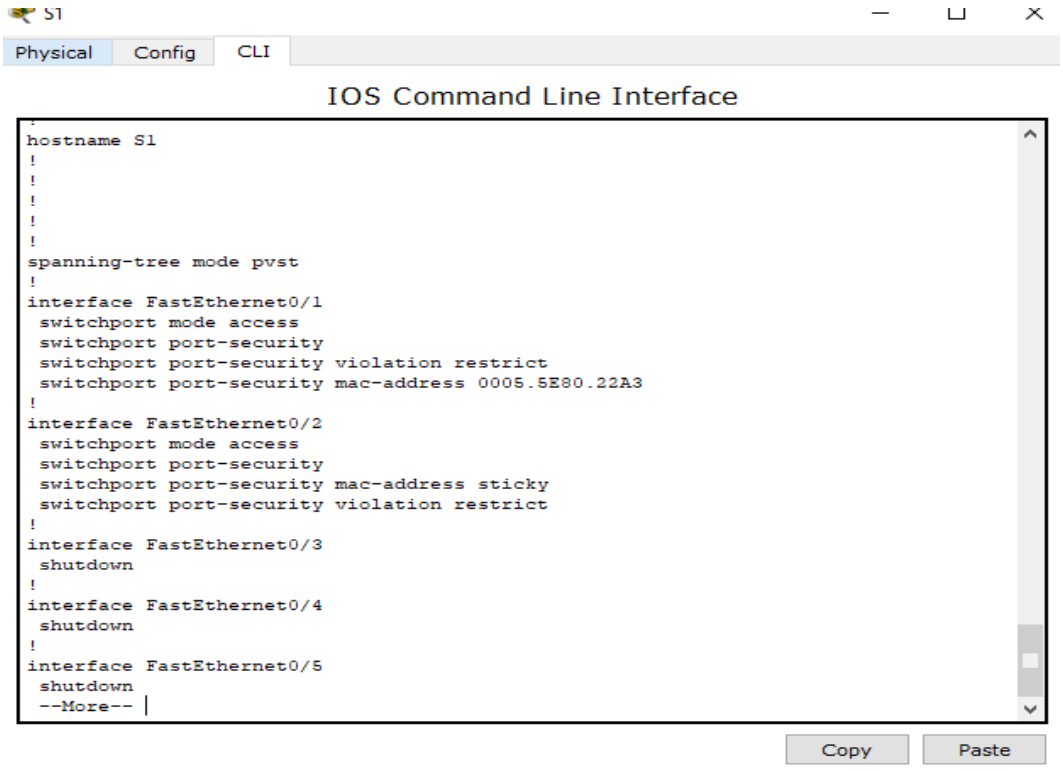

Рисунок 2.2 – Налаштування комутатора в режимі Restrict

У результаті виконання той трафік, що надходив з невідомої адреси буде відкинутий, а також на екран виведуться повідомлення про помилку, та лічильник порушень має збільшуватись. Саме цей режим вважається найоптимальнішим для налаштування мережі в офісах компаній для безпечної роботи.

Конфігурація комутатора з режимом Shutdown:

en

conf t

int fa 0/1 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation shutdown – задаємо режим доступу Shut down.

switchport port-security mac-address 0005.5E80.22A3

exit

int range fa 0/3-24 – вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/1 – перевірка стану порту.

Аналогічні дії потрібно виконати і для fa 0/2:

en

conf t

int fa 0/2 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation shutdown – задаємо режим доступу Shut-down.

switchport port-security mac-address sticky

exit

int range fa 0/3-24 – вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/2 – перевірка стану порту.

Поглянемо на результат за допомогою команди sh run (рис. 2.3).

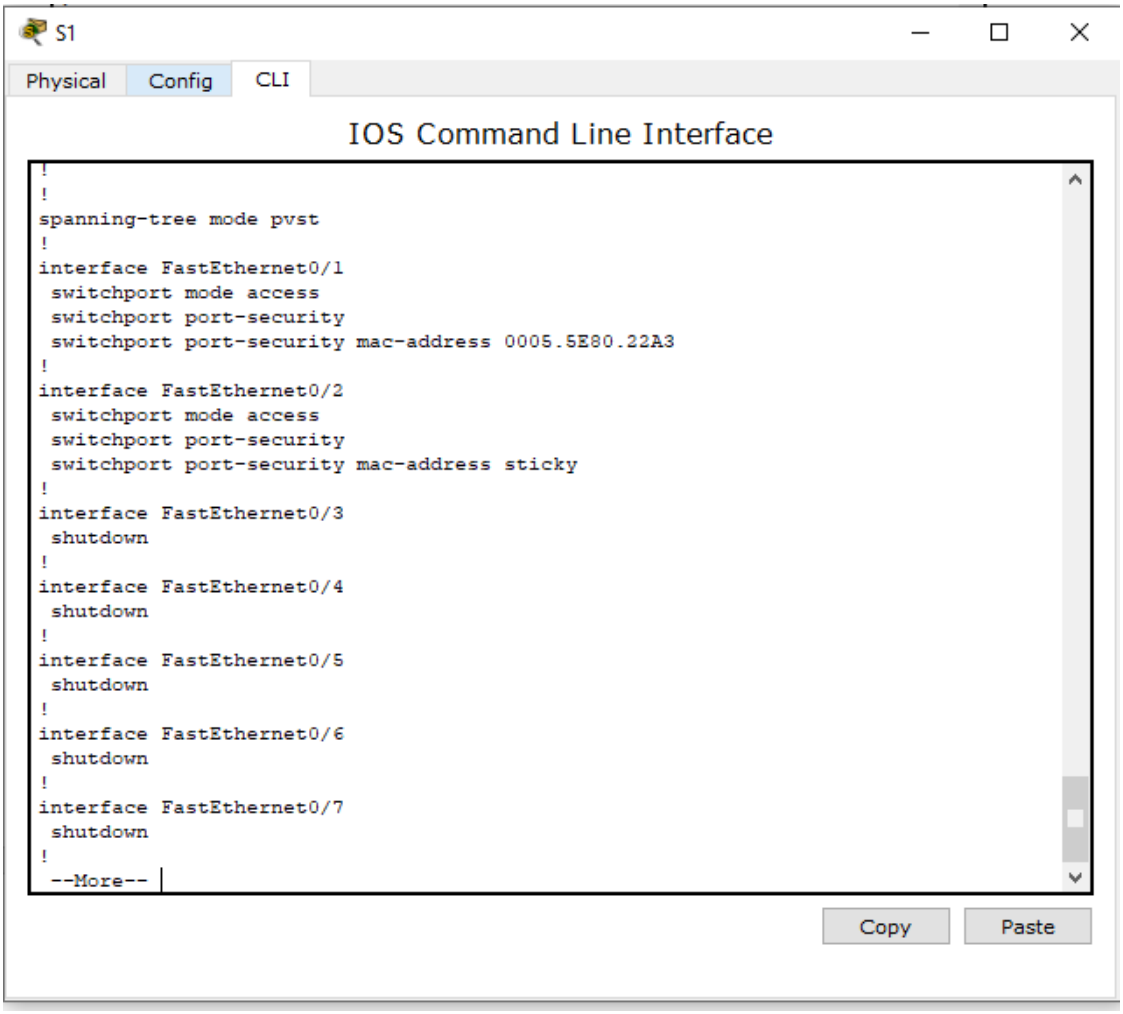

Рисунок 2.3 – Налаштування комутатора в режимі Shutdown

Режим shutdown, робить усе можливе аби негайно зупинити несанкціонований трафік від мережі за допомогою відключення інтерфейсів, а також на екрані будуть виводитись помилки і лічильник порушень буде збільшуватись. Якщо ж ми хочемо вийти з цього режиму то потрібно буде спочатку вимкнути а потім увімкнути інтерфейс за допомогою команд – shutdown та no shutdown.

Конфігурація комутатора з режимом Protect:

en

conf t

int fa 0/1 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation protect – задаємо режим доступу Protect.

switchport port-security mac-address 0005.5E80.22A3

exit

int range fa 0/3-24 – вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/1 – перевірка стану порту.

Аналогічні дії потрібно виконати і для fa 0/2:

en

conf t

int fa 0/2 – обираємо порт для налаштування.

switchport mode access – налаштування режиму інтерфейсу в режимі доступу.

switchport port-security

switchport port-security maximum 1 – максимальна кількість MAC адрес.

switchport port-security violation protect- задаємо режим доступу Protect.

switchport port-security mac-address sticky

exit

int range fa0/3-24 – вимикаємо порти, що не використовуємо.

shutdown

sh port-security interface fa 0/2 – перевірка стану порту.

Поглянемо на результат налаштувань для обох портів за допомогою команди sh run на комутаторі (рис. 2.4).

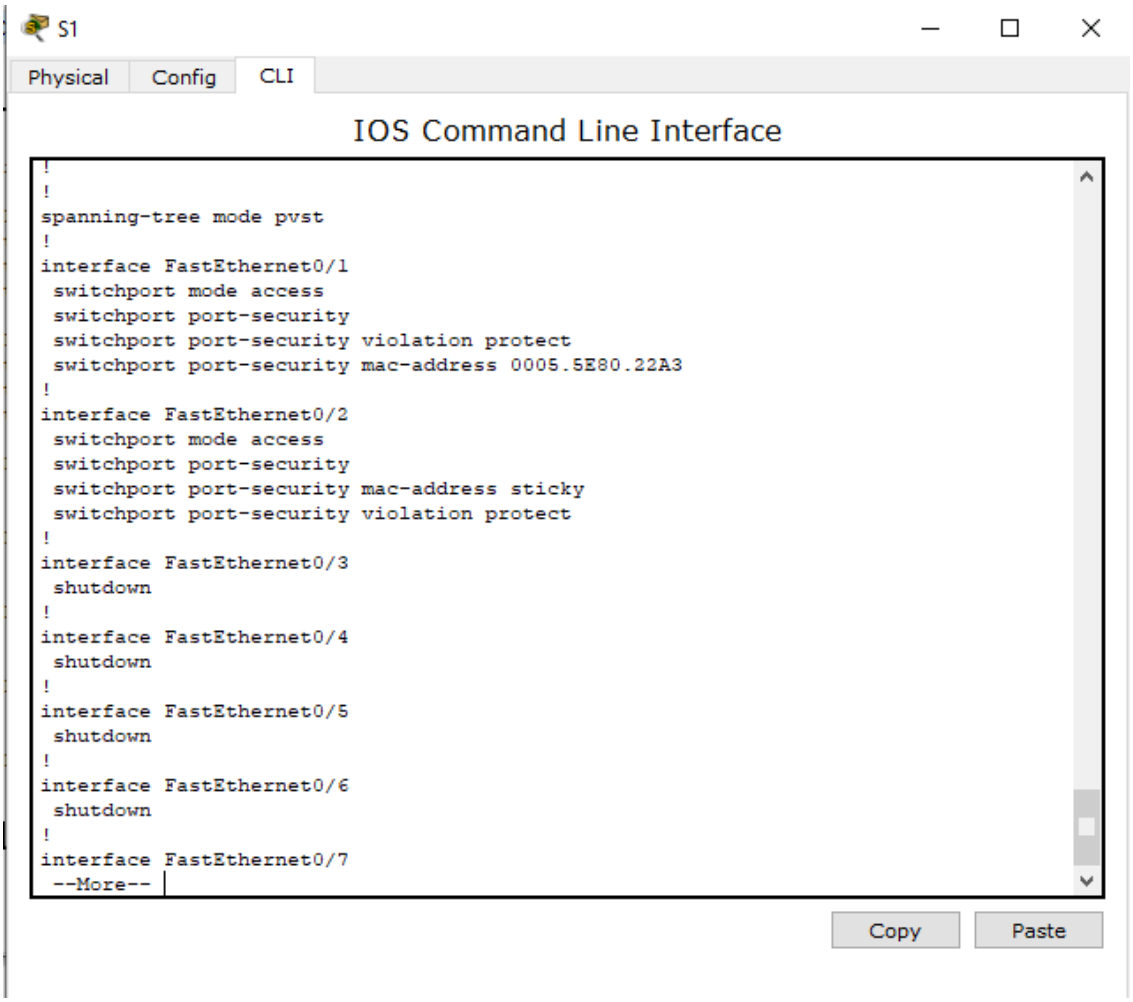

Рисунок 2.4 – Налаштування комутатора в режимі Protect

Режим реагування Protect повинен знищувати також всі пакети, які будуть приходити з несанкціонованого комп'ютера, а також ніякі повідомлення про помилку не будуть надходити з інтерфейсу та не буде збільшено лічильник порушень.

2.2 Конфігурація мережі з функцією безпеки для комутаторів на базі комутаторів Cisco

Cisco Systems – це американська корпорація зі штаб-квартирою у Каліфорнії, яка займається розробкою, виготовленням та реалізацією телекомунікаційного обладнання та інших продуктів та послуг, пов'язаних з цією сферою. Ця велика корпорація відома всьому світу, бо займається виготовленням якісної телекомунікаційної продукції. Їх старт почався ще у 1984 році, завдяки Леонарду Босаку та Сенді Лернеру, вони змогли подолати багатьох конкурентів та стати лідерами на світовому ринку з відомим на весь світ ім'ям. Шлях до світового

визнання був тернистим, бо в цих роках мережа Інтернет ще не набула такої популярності як зараз, і люди не вірили в те, що ця мережа здатна замінити всі відомі на той час способи спілкування та обміном чи передавання інформації. Саме тому спочатку обладнання Cisco не мало великих продажів, але засновники не здавалися, бо знали що як тільки комп'ютери почнуть з'являтись в кожному будинку, їх обладнання стане потрібним кожному[12].

Корпорація виробляє продукти для великих корпорацій (навіть на замовлення) для великого та середнього бізнесу, але також можна використовувати для домашнього користування [13].

Курси телекомунікаційних технологій від Cisco вважаються найбільш престижними, та дають той фундамент знань, який необхідний для повного розуміння телекомунікацій. Ці курси будуть корисними і для нового користувача, що лише починає своє знайомство з мережами, та й для більш досвіченого, що має за мету розширити свої знання.

Саме за допомогою Cisco комутатора та комп'ютерів було змодельовано мережу для середнього офісу з використанням функції безпеки комутаторів для максимального захисту мережі від зловмисників, що можуть заволодіти інформацією.

Для експерименту було використано 2 звичайних комп'ютери та один комутатор, на якому й повинні відбуватись налаштування режимів реагування на небезпеку. Взагалі мережу можна розширити й іншими комп'ютерами, адже до одного комутатора можливо підключити 24 комп'ютери, та на кожному з них налаштувати функцію безпеки з різними режимами та MAC-адресами.

На виконаний експеримент було потрачено близько трьох годин, що є досить великим показником для налаштування мережі в реальному житті. Тобто системний адміністратор має витратити три години робочого часу для налаштування мережі з режимами реагування, але це займає майже половину робочого дня, що не допустимо на великих підприємствах, де кожна хвилина має цінуватись. Тому аби знизити цей час, було розроблено веб-орієнтований додаток,

який легко запускати в вікні браузера. Та ввівши всі необхідні значення, необхідні конфігураційні будуть згенеровані за декілька хвилин, а потім можливо їх скопіювати до інтерфейсів комутаторів.

Веб-додаток, розроблений таким чином, що підтримує всі відомі браузери, та має змогу працювати в оффлайн-режимі. Це означає, що для запуску додатку не потрібно мати доступ до мережі Інтернет. Тим самим, знижуючи ризик атак, на додаток з мережі. Необхідно лише встановити програмний додаток на робочу станцію та запустити.

2.3 Розробка веб-орієнтованої системи за допомогою фреймворку Vue Js

Нині відомо значну кількість фреймворків, що були розроблені мовою Java Script, для полегшення роботи програміста.

Було вирішено розробити інтерфейс за допомогою фреймворку Vue Js, для того аби мінімізувати написання однотипного коду, та позбутись рутинності. Фреймворк задовольняє всі необхідні потреби для інтерфейсу, адже містить вже в собі необхідні інструменти та бібліотеки.

Розробленням цього фреймворку займався Іван Ю, що відкрив світу Vue Js вже у 2014 році в Китаї. А потім був випущений Vue Js 2 в 2016 році. Останній реліз зробив фреймворк ще кращим.Він повністю безкоштовний, та його можна завантажити з офіційного сайту для Java Script. Фреймворк здобув визнаність у таких великих корпорацій, як Google, Facebook та Xiomi. Вже велика кількість веб-додатків розроблена завдяки цьому фреймворку. І це зовсім не випадково, адже фреймворк має значні переваги в продуктивності вашого додатку, підтримує серверний рендерінг, а також використовує віртуальний DOM, підтримує Type Script для покращення якості коду та робить його легшим для розуміння.

Також до унікальних властивостей Vue JS можна віднести що ядро бібліотеки має лише рівень перегляду, а тому можливе інтегрування до інших бібліотек.

Маршрутизація сторінок виконується за допомогою vue-router. І за часом оброблення функції Vue JS займає перше місце серед схожих фреймворків. Це є

його головною особливістю, яка виділяє його серед інших. Також фреймворк займає досить незначну пам’ять на комп’ютері, що є важливо для програміста. Можливість встановлення є досить різноманітною, і має декілька способів:

- встановлення за допомогою Node Package Manager (MPN).

Для цього нам потрібно встановити на комп’ютер Node.js, а потім ми можемо створити свій додаток на Vue Js, за допомогою команди mpn install vue, а для створення проекту потрібно виконати vue create.

- встановлення за допомогою Conten Delivery Network (CDN).

Для цього потрібно скопіювати посилання з офіційного сайту Vue Js до нашого проекту для підключення.

Для мого проекту є важливою можливість використання html та css з JS разом, так ми зможемо контролювати кожен з них окремо в одному місці. Це означає що не потрібно більше створювати декілька файлів, які можуть бути втрачені. Тепер все можливо тримати в одному файлі.

На початку роботи нам потрібно мати лише Node Package Manager, який можна встановити безкоштовно з офіційного сайту, та виконавши конфігураційні команди, ми отримаємо пустий проект. Також в разі інтеграції мого проекту до іншого це буде легкою задачею, адже цей фреймворк підтримує інтегрування та гнучкість.

В ході виконання роботи були використані такі концепції:

- конструктор
- компоненти
- зміни
- бібліотеки

Конструктор повинен містити в собі певний набір компонентів, що використовується додатком для функцій та бібліотек, які задані за замовчуванням. Директиви використовуються для html розмітки тексту. Тобто, ця концепція відповідає за зовнішній вигляд веб-орієнтованої системи. А зміни будуть використані для анімацій елементів коду, до них можна віднести:

- використання інших бібліотек, що не включені до фреймворку для анімацій;
- використання вже заданої бібліотеки;
- використання написаного коду на основі Java Script;
- використання Java Script бібліотек.

Для написання коду для логіки роботи веб-орієнтованого додатку та підтримки валідації для негативних випадків потрібно буде запустити сервер для компіляції нашого коду та визначення критичних помилок в роботі додатку. Саме тому й було обрано цей фреймворк, через його зручність та простоту в роботі.

На розробку веб-додатку було витрачено близько 30 годин, що можна вважати досить складною роботою. Налаштування стабільної версії додатку зайняло ще годину. Але тепер його можна використовувати на будь-якому апаратному забезпеченні та операційній системі, що робить його крос-платформенним.

Особливістю додатку є те, що він має можливість запуску через комп’ютер, ноутбук, та навіть телефон. Тобто він підтримує функцію портативності для додатку.

3 ПРОГРАМНО-ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ НАЛАШТУВАННЯ ФУНКЦІЇ БЕЗПЕКИ КОМУТАТОРІВ

3.1 Розробка графічного інтерфейсу налаштування функції безпеки комутаторів

Мережа з налаштуванням безпеки комутатора була зроблена в симуляторі Cisco Packet Tracer з використанням режимів реагування: protect, restrict, shutdown, MAC- адресами як статичними так і sticky, і проведено відключення усіх незалучених портів для інтерфейсів. Та ця конфігурація була відтворена на справжньому обладнанні. Отже, було виявлено головний недолік для Cisco Packet Tracer – це те, що немає графічного інтерфейсу для швидкого налаштування. Цей недолік робить процес налаштування безпеки комутаторів досить тривалим. Тому розроблена веб-орієнтована система робить цей процес легким та швидким, що є його перевагою.

Програма була розроблена за допомогою Java Script на базі фреймворку View JS.

Інтерфейс був зроблений доступно, та не потребує спеціальної підготовки.

Проект запускається через будь-який браузер локально. Але можливо помістити його на хост для загального користування.

При відкриванні сторінки ви побачите інтерфейс, що містить в собі 6 полів для вводу інформації: порт, кількість MAC-адрес, MAC-адреса, порти на відключення, стан порту, та sticky MAC-адреса. А також є 1 поле для вибору режиму реагування на небезпеку: protect, restrict, shutdown. Генерації конфігурації використовуємо кнопку – Generate, а для копіювання виведеної інформації – Copy. Варто помітити, що є 3 блоки: перший блок відповідає за ввід інформації, другий за приклад мережі в якій відбувається налаштування, та третій блок – це консоль, де буде виводитися уся інформація з заповненими відповідними значеннями полями.

Програма за замовчуванням немає жодного заповненого поля, та містить кількість MAC – адрес одну. Консоль теж немає має бути порожньою. Отже, для запуску програми потрібно лише заповнити поля як на рис. 3.1.

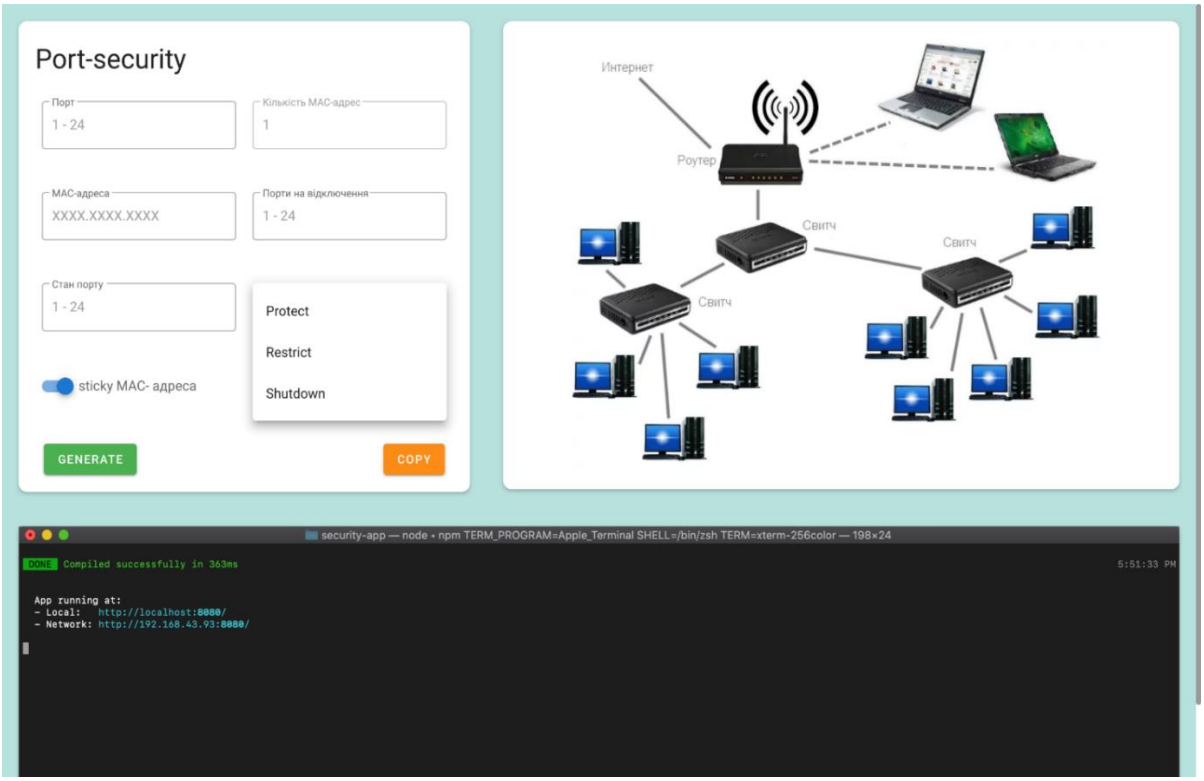

Рисунок 3.1 – Інтерфейс веб-орієнтованого графічного інтерфейсу

Адміністратор має відкрити додаток в браузері для подальшого заповнення усіх необхідних полів.

Заповнити поля необхідно відповідно до його потреби в використовуваних MAC- адресах, та режимах реагування, що використовують на підприємстві. Режими реагування потрібно обирати орієнтуючись на потреби.

Як відомо є три режими реагування на небезпеку: restrict, protect, shutdown. В роботі ми роздивимось усі три режими реагування. Також випадки з використанням як MAC- адреси так і sticky MAC- адреси для декількох портів.

На рис. 3.2 поданий приклад конфігурації зі sticky MAC- адресою та режимом реагування shutdown, а також портами на відключення, які не мають брати участі.

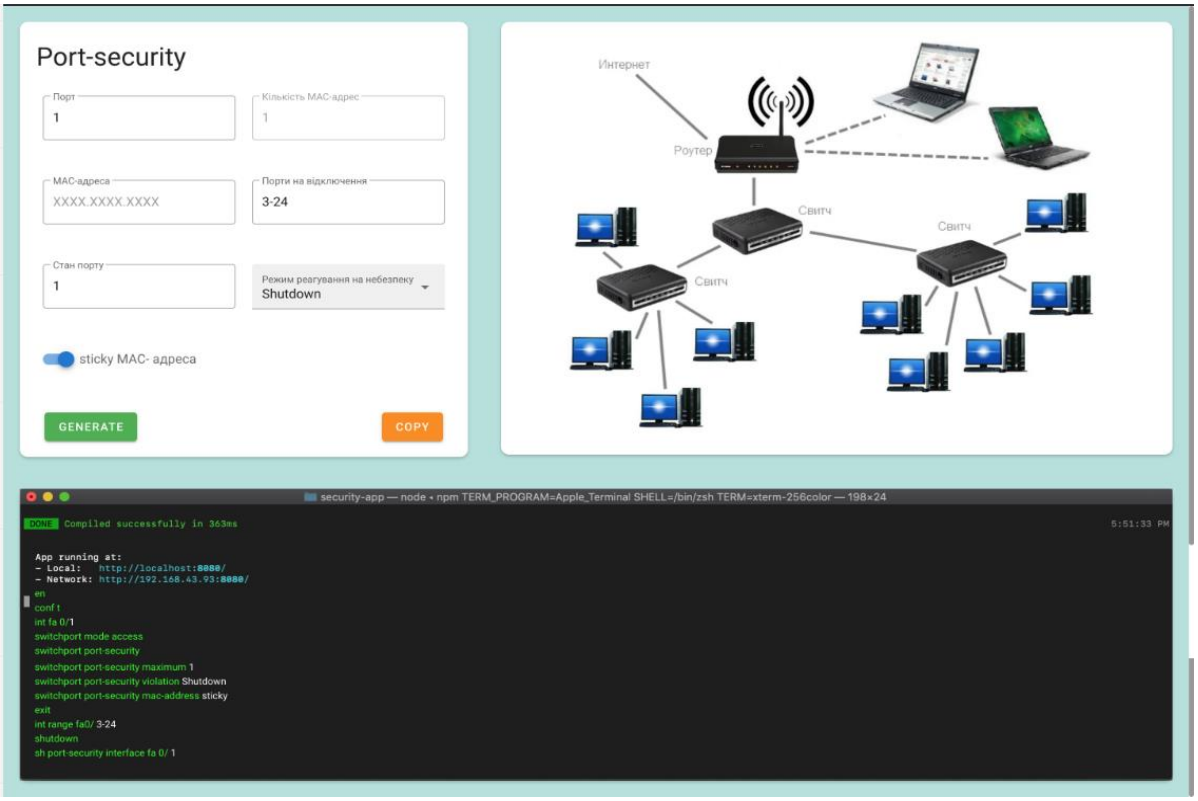

Рисунок 3.2 – Налаштування з режимом реагування Shutdown та Sticky MAC- адресою

Конфігурація що була виведена знаходиться в консолі та позначена зеленим кольором (рис. 3.3):

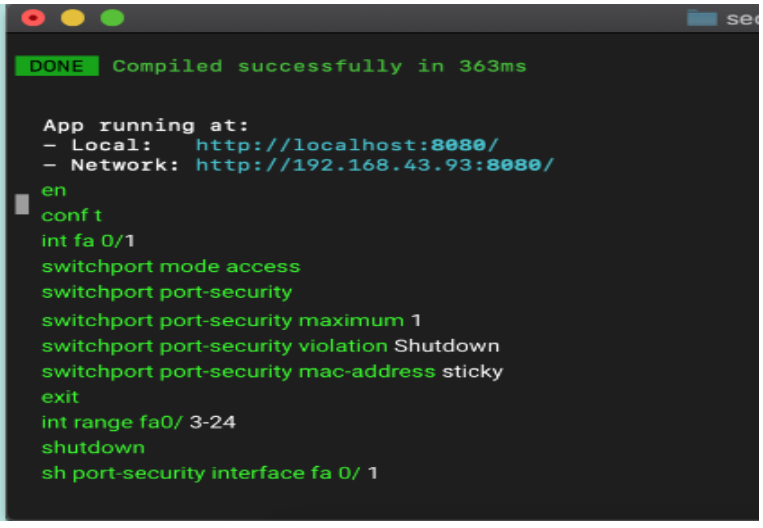

Рисунок 3.3 –Конфігурація в консолі з режимом реагування Shutdown та Sticky MAC- адресою

Також можна налаштувати статичну MAC-адресу з Restrict режимом реагувння (рис. 3.4).

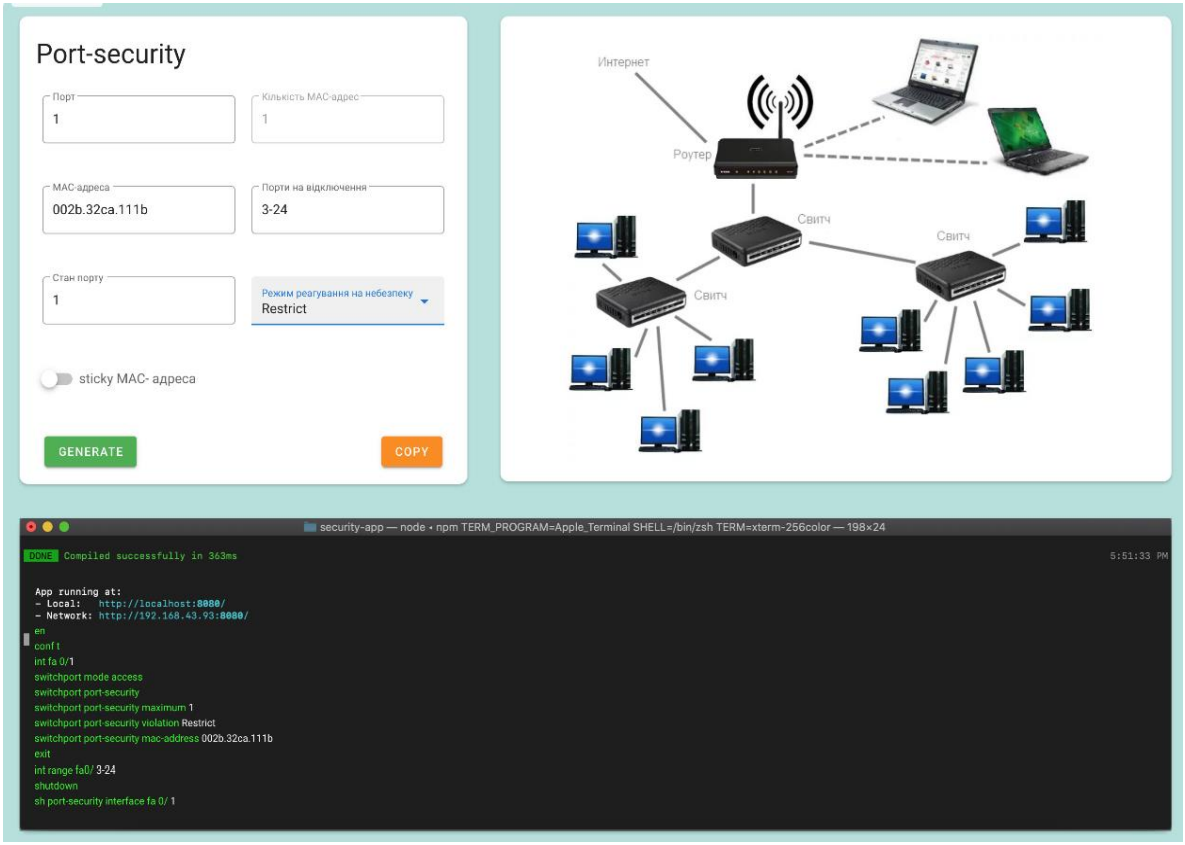

Рисунок 3.4 – Налаштування з режимом реагування Restrict та статичною MAC- адресою

Конфігурація для режиму Restrict була також виведена в консолі та визначена зеленим кольором (рис. 3.5):

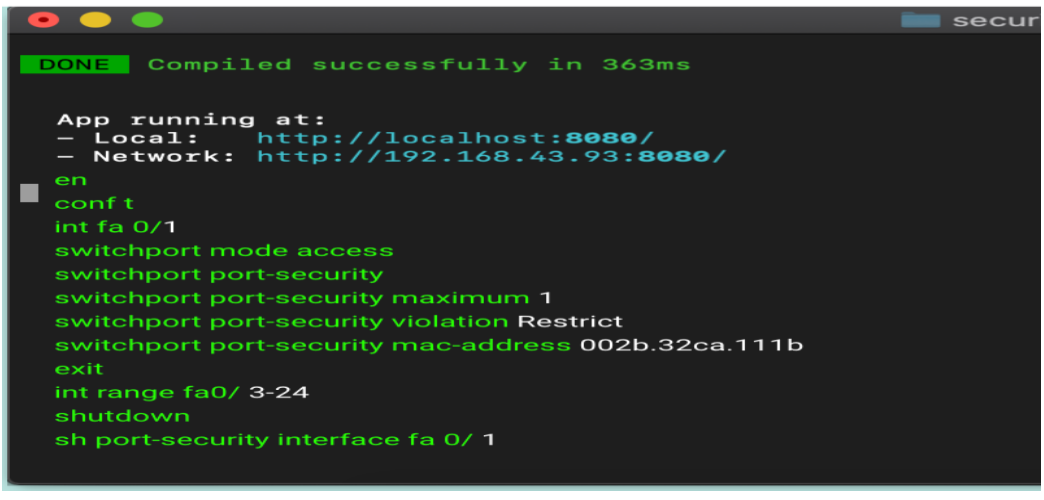

Рисунок 3.5 –Конфігурація в консолі з режимом реагування Restrict та статичною MAC- адресою

Та останній варіант налаштування це статична MAC-адреса з Protect режимом реагування (рис. 3.6).

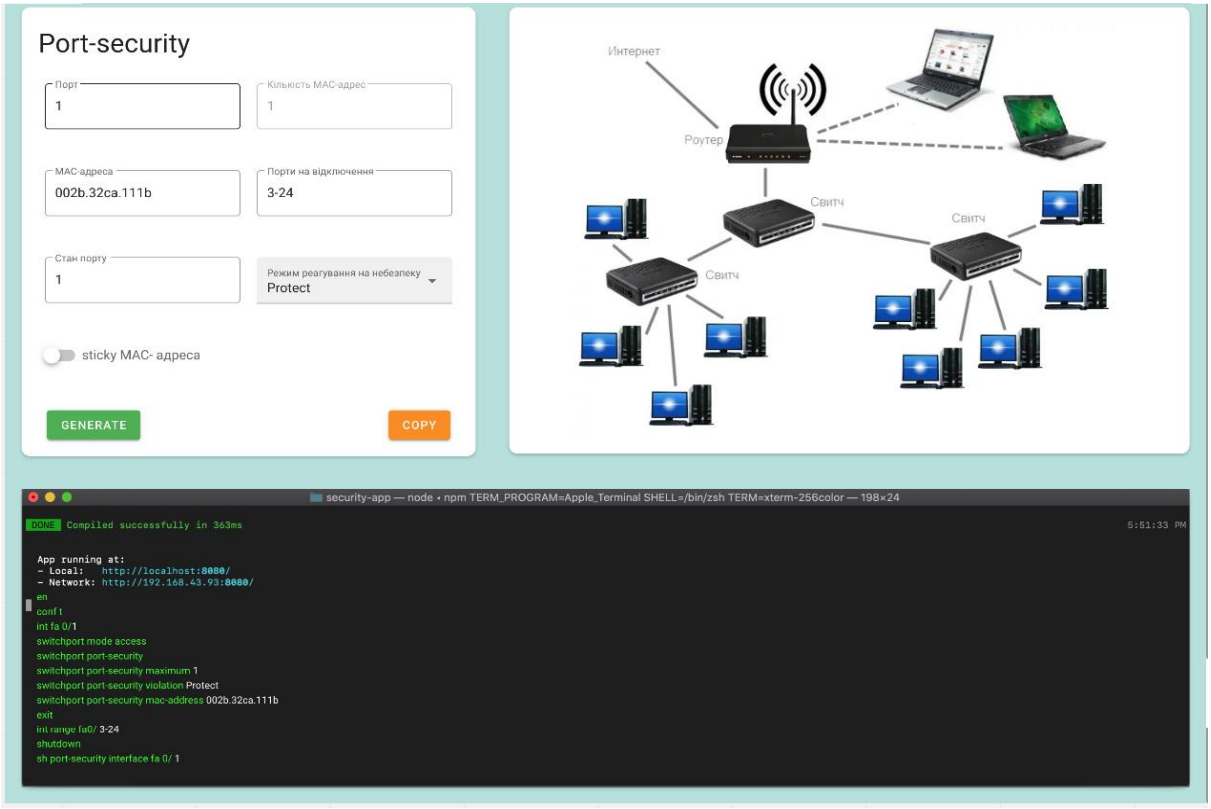

Рисунок 3.6 – Налаштування з режимом реагування Protect та статичною MAC- адресою

Варто звернути увагу на білий колір та зелений для команд конфігурації, що знаходиться на консолі. Зеленим кольором позначені команди, які не повинні змінюватись та є однаковими для всіх режимів. Білим кольором позначені введені користувачем данні, що можуть змінюватись відповідно до вимог. Синій колір даних не несе ніякої цінності, адже це лише час, коли були введені данні.

Малюнок має допомагати адміністратору краще зрозуміти для якої мережі будуть зроблені налаштування, адже на ньому зображений приклад мережі.

Що стосується режиму реагування Protect зі статичною MAC- адресою, то консоль має такий же вигляд, як і для інших двох режимів (рис. 3.7)

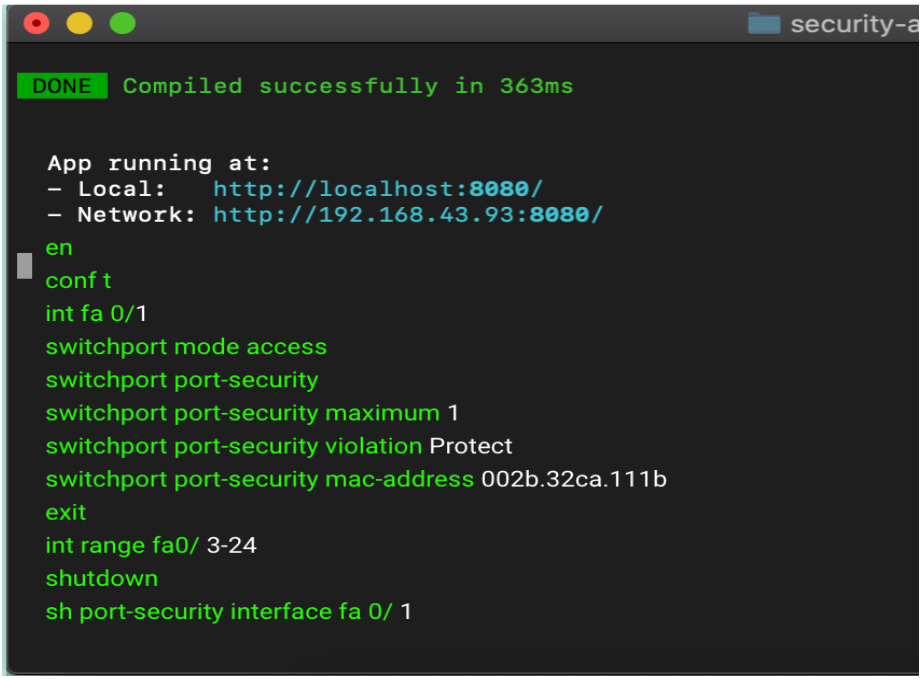

Рисунок 3.7 – Конфігурація в консолі з режимом реагування Protect та статичною MAC- адресою

Також графічний інтерфейс має валідацію, що повинна перевіряти вірність введених даних користувачем. Та виводити змістовне повідомлення про помилку, що сталась. З цього повідомлення користувачу буде легко зрозуміти що саме не так, та виправити введені ним данні. Кількість спроб введення параметрів є необмеженим, і тому введення буде відбуватись до тих пір, поки помилка не зникне і система поверне результат. Конфігурація не буде генеруватись в разі помилки.

До валідації відноситься: перевірка на довжину символів, на кількість введених адрес, довжину введеного MAC-адресу, та неможливість введення одночасно sticky та статичного адресу (рис. 3.8).

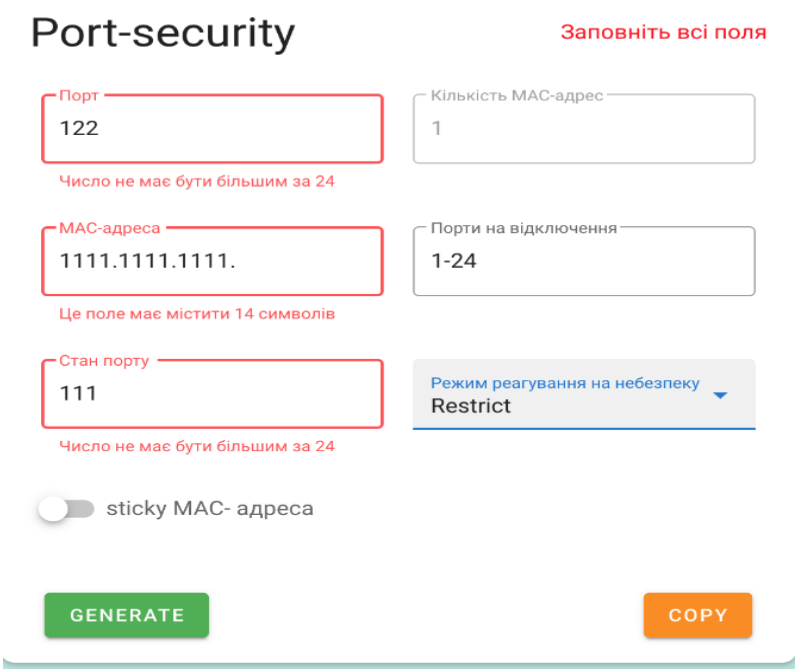

Рисунок 3.8 – Валідація полів вводу

Якщо, адміністратор спробує натиснути кнопку Generate та не заповнить усіх полів, то він також отримає повідомлення з проханнями ввести всі поля (рис. 3.9).

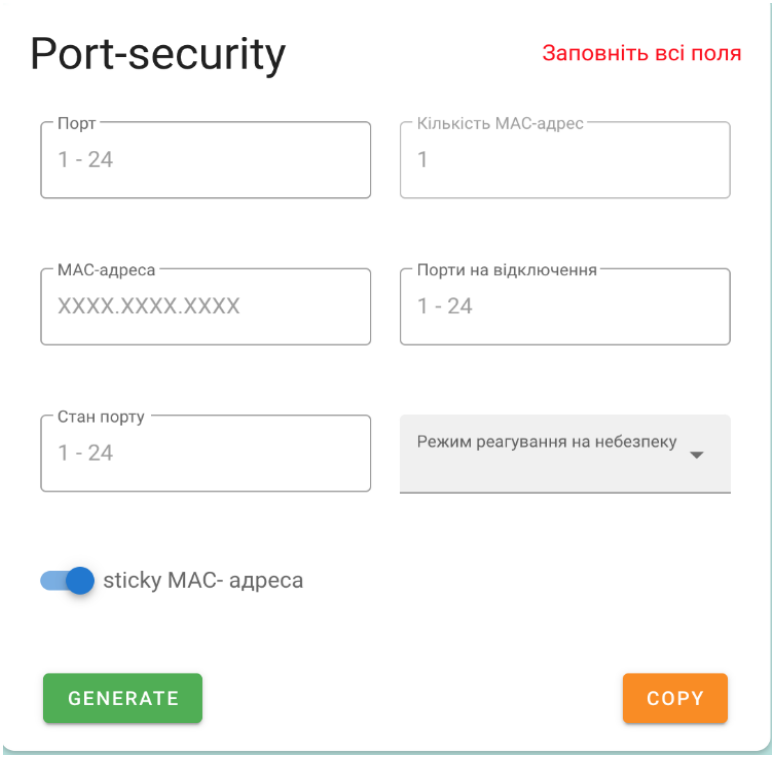

Рисунок 3.9 – Валідація порожніх полів вводу

Щоб налаштувати цю конфігурацію на комутаторі потрібно лише обрати режим реагування та заповнити всі необхідні поля та натиснути кнопку Generate,

а потім Copy щоб зберегти інформацію у буфер обміну та вставити надалі його у справжній комутатор.

Обов’язково для перевірки необхідно виконувати команду sh run, яка відображає всі налаштування на портах комутатора, та які порти їх мають а які ні. Це досить зручна команда, що допомагає швидко виявити проблему та виправити її в короткий час.

У даній роботі будемо використовувати команду «sh port-security interface fa», яка показує стан налаштування саме безпеки порту, де добре видно стан порту, ввімкнений він чи ні, режим реагування. Цю команду потрібно виконувати лише у разі необхідності, якщо потрібно бути впевненим в вірності введених даних.

3.2 Тестування веб-орієнтованої інформаційної системи в симуляторі Cisco Packet Tracer та на реальному обладнанні Cisco

Тепер потрібно перевірити чи працює веб-додаток так як потрібно. Для цього потрібно зробити тест за допомогою Cisco Packet Tracer та справжнього обладнання. Це дозволить нам зрозуміти, чи дійсно веб-система відповідає всім поставленим вимогам та працює злагоджено та без дефектів.

Отже, розпочнемо перевірку з налаштування безпеки комутаторів, яка полягає в обмеженні вхідного трафіку за рахунок саме обмеження MAC- адрес, які можуть посилати трафік через цей порт.

Для початку роботи відкриваємо в будь-якому браузері веб-додаток, та задаємо функцію безпеки для портів 0/1 та 0/2 інтерфейсу FastEthernet з режимом реагування на небезпеку, відповідно до вимог який би дозволяв залишити порти ввімкненими, але всі чужі пакети відкидались би – це Protect режим реагування на небезпеку та вказуємо лише один пристрій в якості максимуму для доступу до портів. Також необхідно вказати MAC –адреси для двох портів, та діапазон портів, що не будуть використовуватись або можна налаштувати sticky MAC- адресу, для якої не потрібно вказувати MAC- адресу . Для того, щоб дізнатись стан порту необхідно також ввести буде цей порт.

Перевіримо роботу режиму реагування – Restrict, який дозволяє в разі небезпеки залишити порти ввімкненими, але всі чужі пакети відкидались би при цьому. За допомогою графічного інтерфейсу тепер потрібно заповнити по черзі кожен порт.

Почнемо з 0/1 порту, для якого заповнимо поля для вводу Порт, Кількість MAC- адрес, Sticky MAC- адресу, Порти на відключення, відповідний Режим реагування на небезпеку та Стан порту (рис. 3.10).

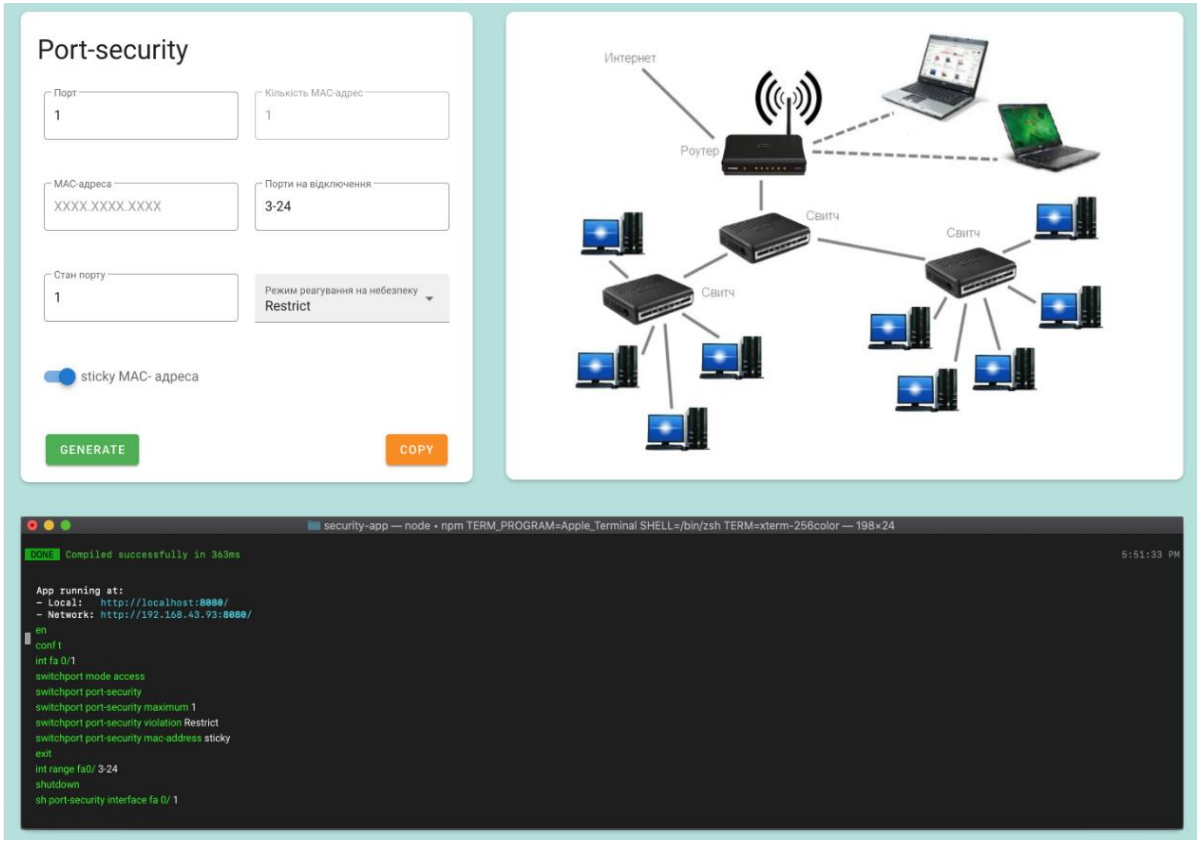

Рисунок 3.10 – Налаштування порту 0/1 з Restrict для комутатора

Тепер тиснемо на кнопку «Generate» і бачимо що код з налаштуванням був згенерований (рис. 3.11).

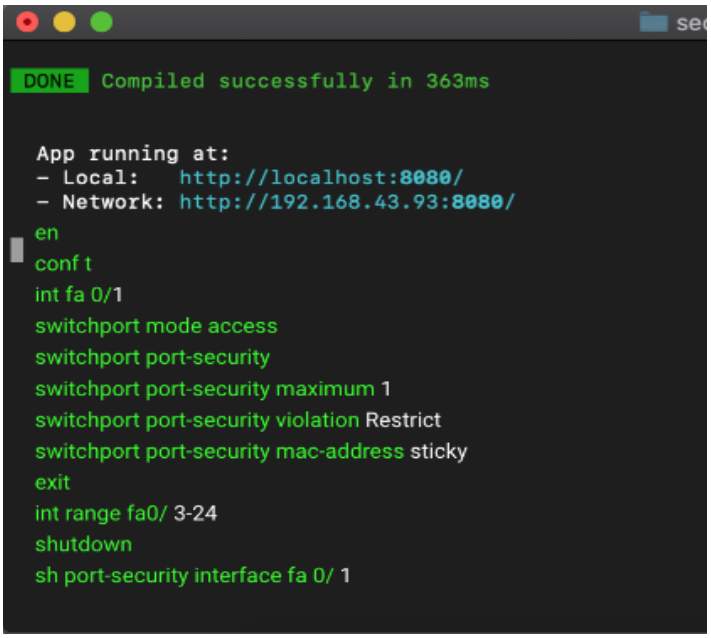

Рисунок 3.11–Код налаштування порту 0/1 з Restrict для комутатора

Як видно, згенеровані команди відповідають всім необхідним командам, що потрібні для налаштування. Білим кольором позначені значення, що мають змінюватись відповідно до введених параметрів, а зеленим позначені команди, що залишаються однаковими та не залежать від випадку.

Щоб застосувати конфігураційні налаштування до комутатора, необхідно його скопіювати за допомогою кнопки «Copy» та вставити в консоль (рис. 3.12).

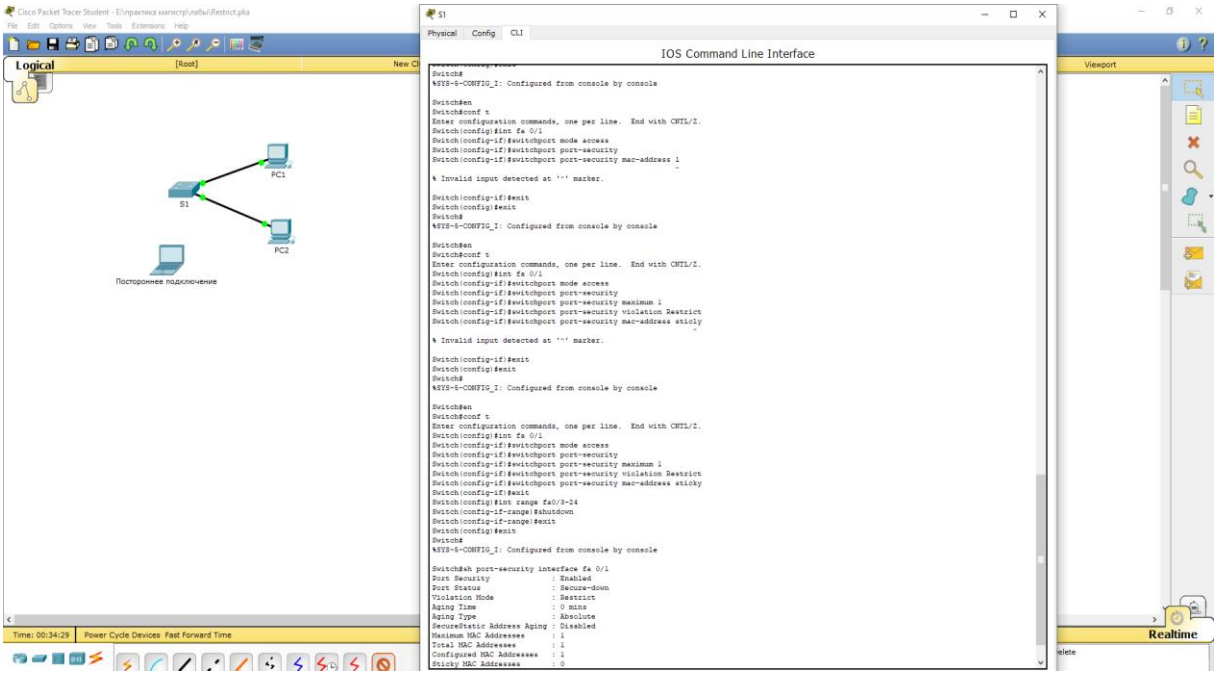

Рисунок 3.12 – Вікно налаштувань роутера для 0/1 порту

З рисунку 3.12 добре видно, що команди злагоджено відпрацювали та не виникло ніяких помилок, адже про це говорить статус порту, який є ввімкненим та на ньому встановлений режим реагування Restrict зі sticky MAC-адресою.

Розглянемо рисунок ближче, де видно що команди відпрацювали без помилок в симуляторі (рис. 3.13).

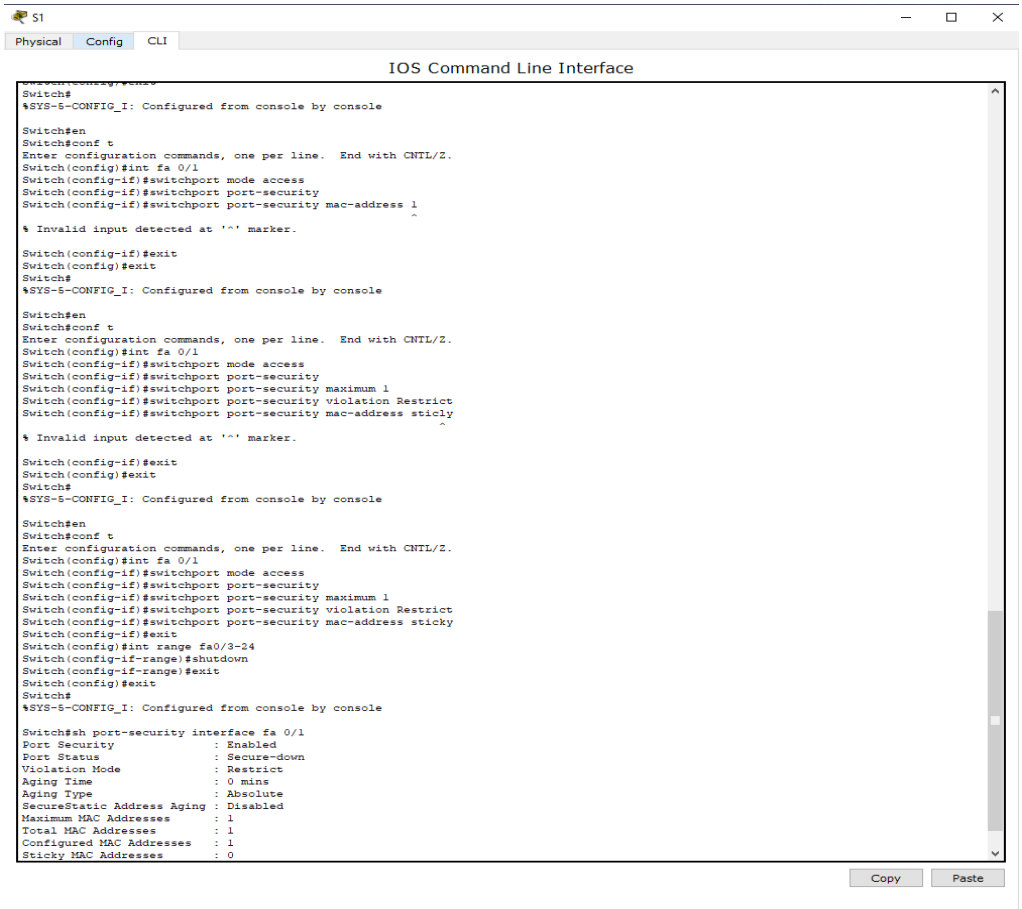

Рисунок 3.13 – Вікно налаштувань роутера для 0/1 порту (2)

Тепер виконаємо аналогічні дії і для другого комп’ютера з іншим вже портом та MAC- адресою (рис. 3.14).

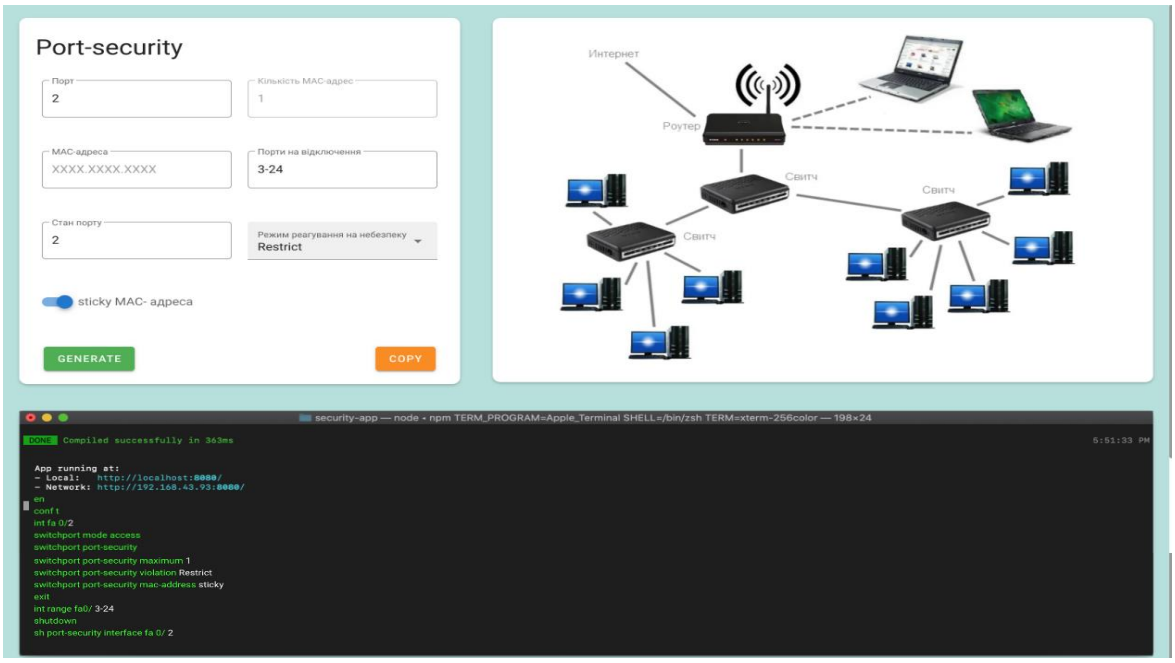

Рисунок 3.14 – Вікно налаштувань роутера для 0/2 порту

Якщо уважно придивитись, то видно, що команди такі ж як і для порту 0/1, що є вірною поведінкою системи (рис. 3.15).

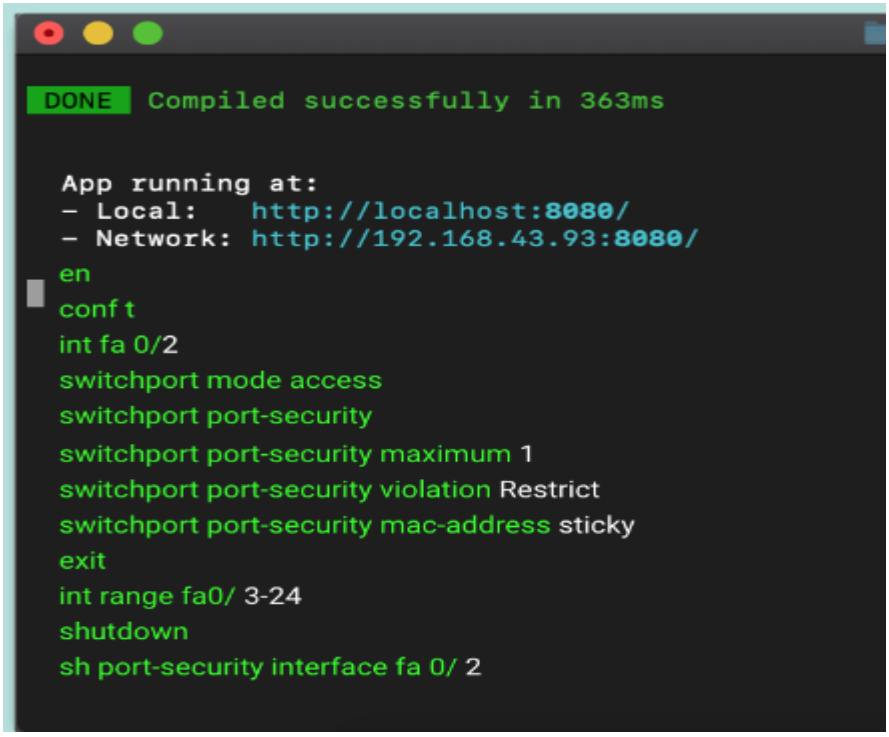

Рисунок 3.15 – Код налаштування порту 0/2 з Restrict для комутатора

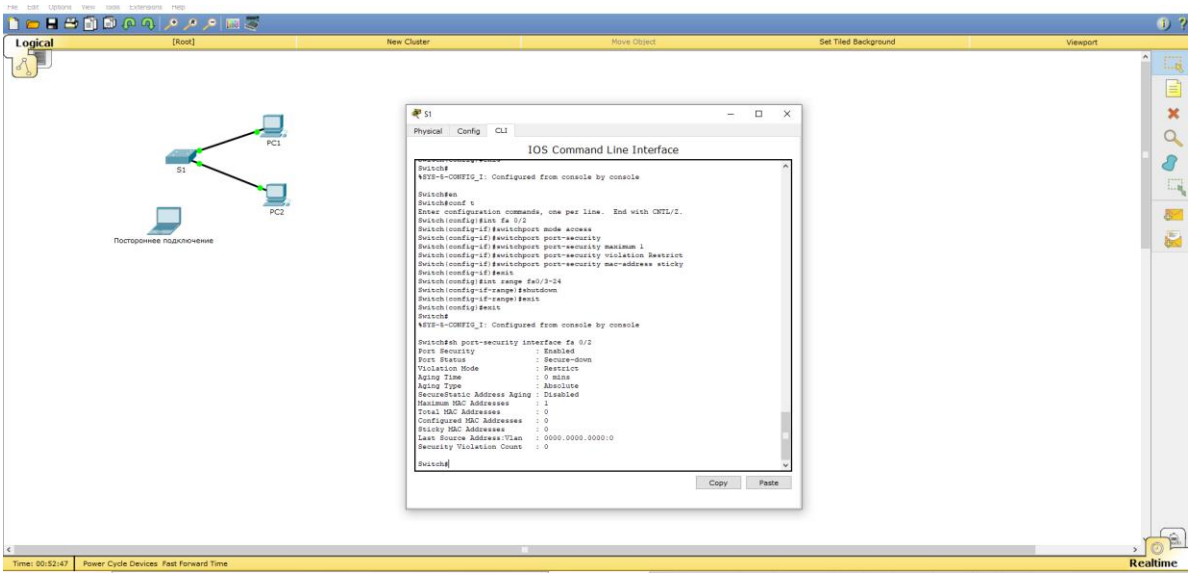

Рисунок 3.16 – Вікно налаштувань роутера для 0/2 порту

Слід також ближче оглянути налаштування комутатора, та переконатись що не виникло помилок (рис. 3.17).

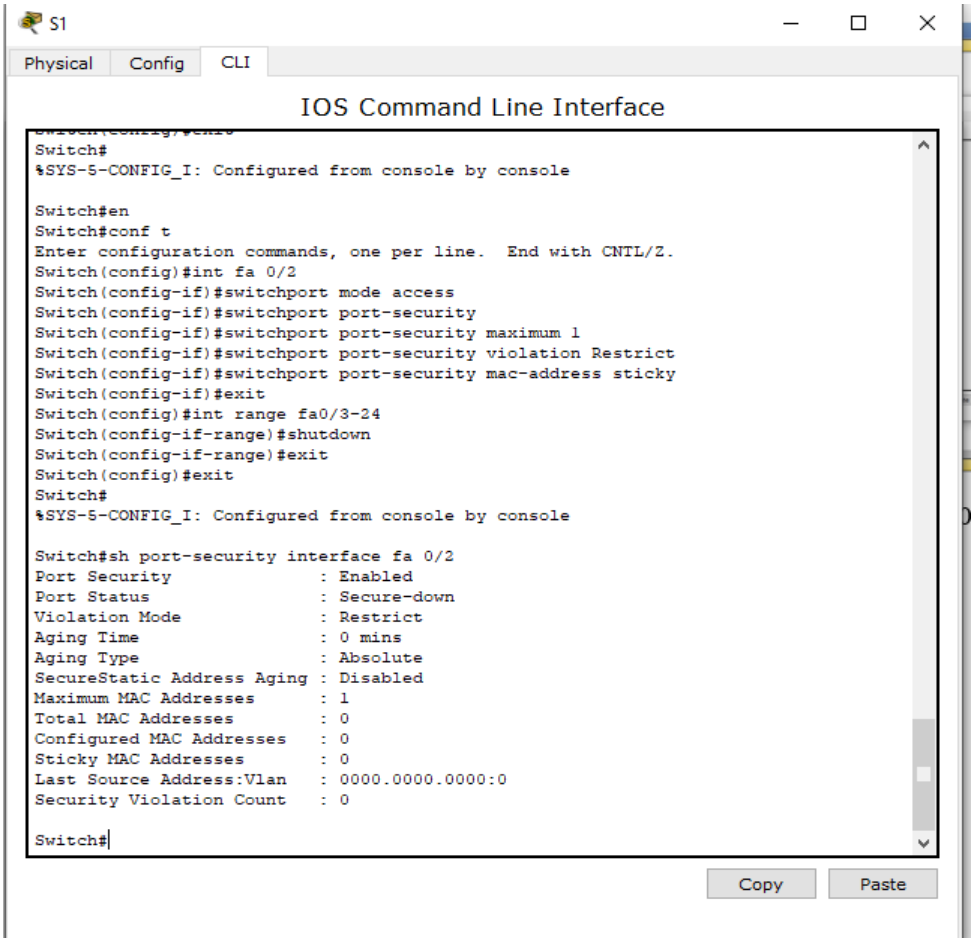

Рисунок 3.17 – Вікно налаштувань роутера для 0/2 порту (2)

Добре видно, що ніяких помилок не виникло та команди відпрацювали належним чином, а тому можемо продовжувати.

Також команда «sh port-security interface fa 0/2» показує стан порту, та налаштування що були виконані. Головне є те що порт є ввімкненим та режим реагування відповідає бажаному.

Тепер переконаємось що налаштування були виконані вірно, для цього необхідно виконати команду «show run» (рис. 3.18).

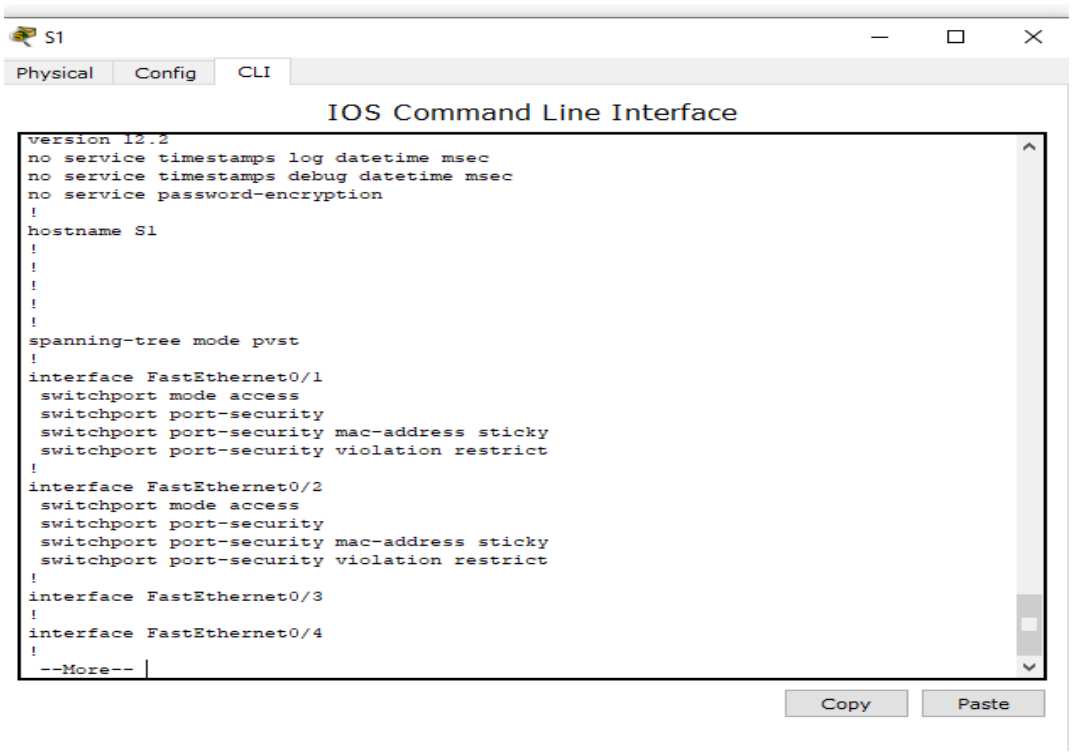

Рисунок 3.18 – Результат виконання команди «show run»

Тепер розглянемо ситуацію, коли до нашої мережі був підключений зловмисник. Спочатку розглянемо, що пінг запит йде від комп’ютера PC1 до PC2 (рис. 3.19).

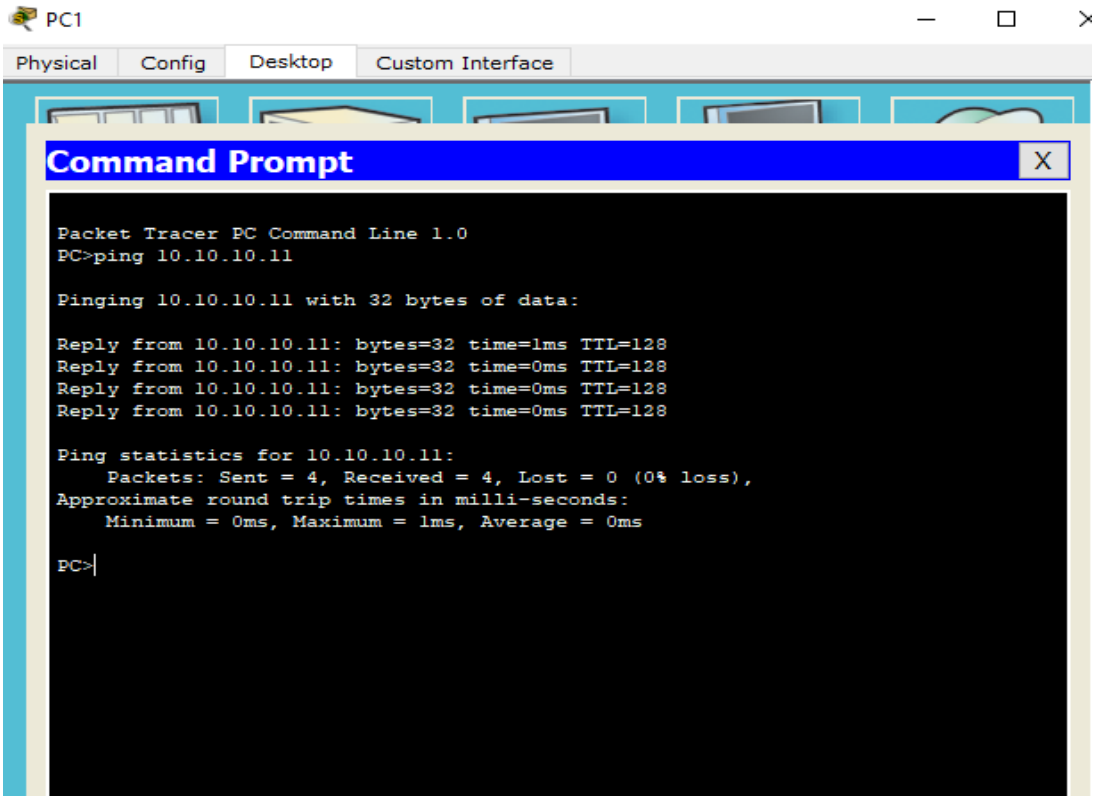

Рисунок 3.19 – Результат виконання пінгування комп’ютера

Отже, роздивимось випадок коли до мережі, яка була захищена за допомогою налаштування безпеки порту. Зловмисник під’єднався до мережі за допомогою ноутбука, з метою заволодіння цінною інформацією та зараженням комп’ютера вірусом, що несе в собі небезпеку для інформації. Ноутбук був з’єднаний до невикористаного порту комутатора. Це з’єднання горить червоним кольором, в той час як санкціоновані з’єднання зеленим (рис. 3.20).

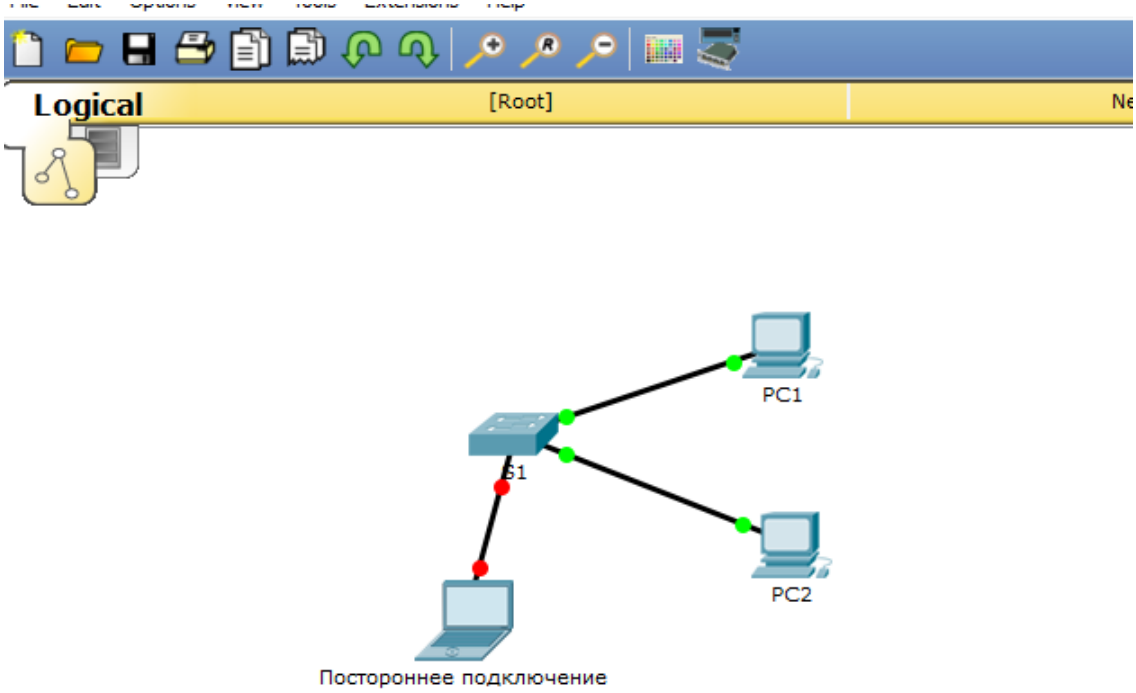

Рисунок 3.20 –Несанкціоноване з’єднання зловмисника з ноутбуком

Ноутбук, що несе в собі пряму небезпеку для нашої мережі не може відправляти запити до наших комп’ютерів.

Далі потрібно вимкнути PC2 та до його порту підключити ноутбук зловмисника, та переконатися, що він не може надсилати єхо-запити, а для цього потрібно знову виконати пінгування ноутбука до PC1 (рис. 3.21).

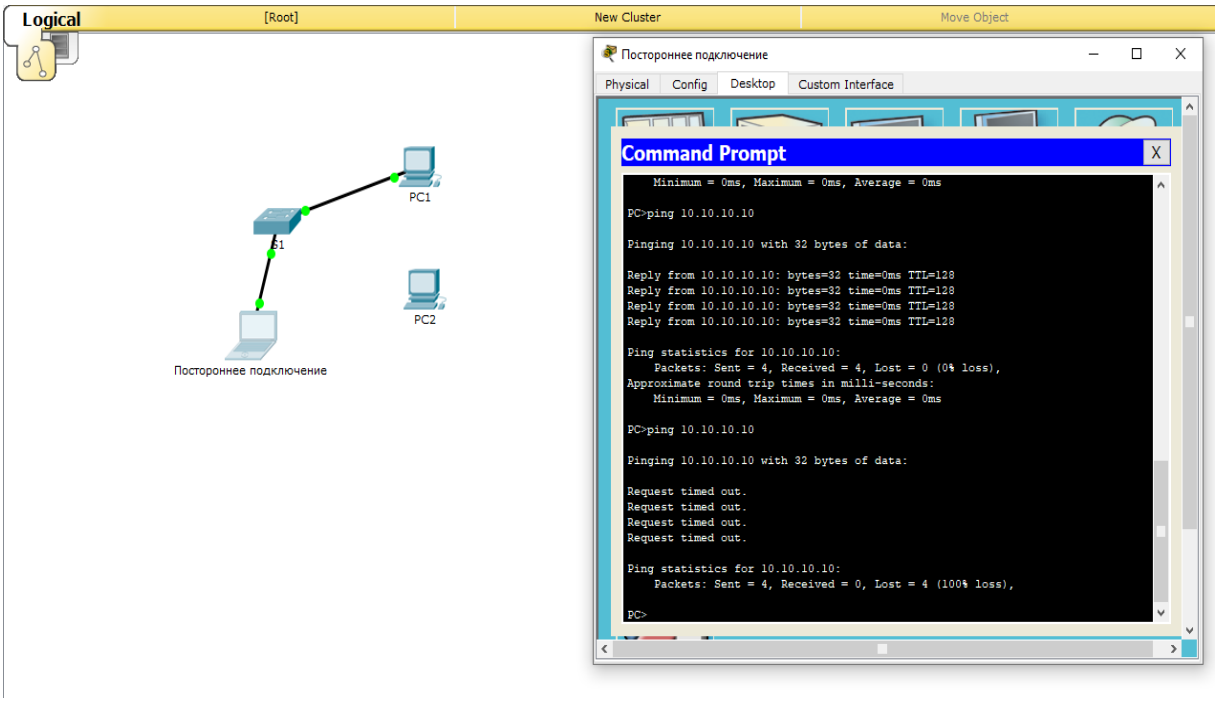

Рисунок 3.21 – З’єднання ноутбука через порт комп’ютера PC2

Переконаємось, що вірно було налаштований режим реагування, а для цього потрібно вивести на екран порушення режиму безпеки (рис. 3.22).

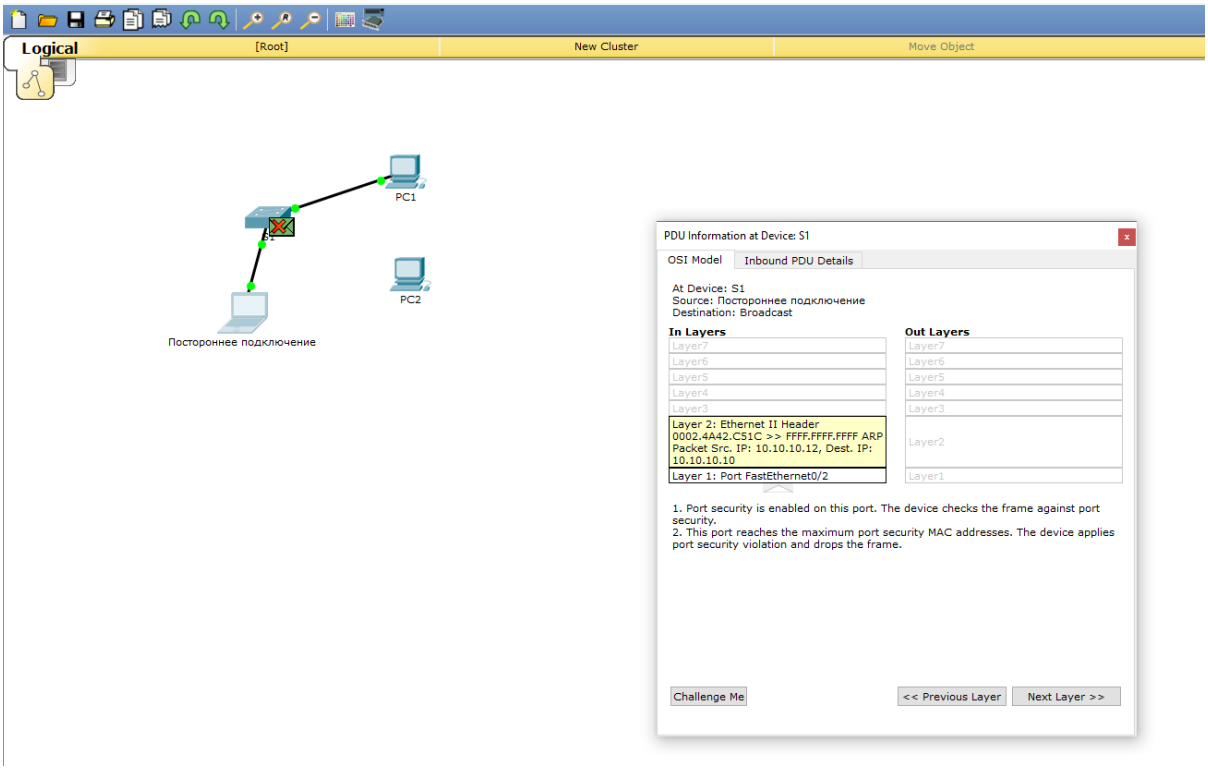

Рисунок 3.22 – Повідомлення про режим порушення безпеки

Після виконаного досліду, потрібно знову повернути комп’ютер PC2 до порту а ноутбук відключити від комутатора. Щоб переконатись, що з’єднання

було відновлено необхідно знову відправити ехо-запит с комп’ютера PC1 до PC2 (рис. 3.23).

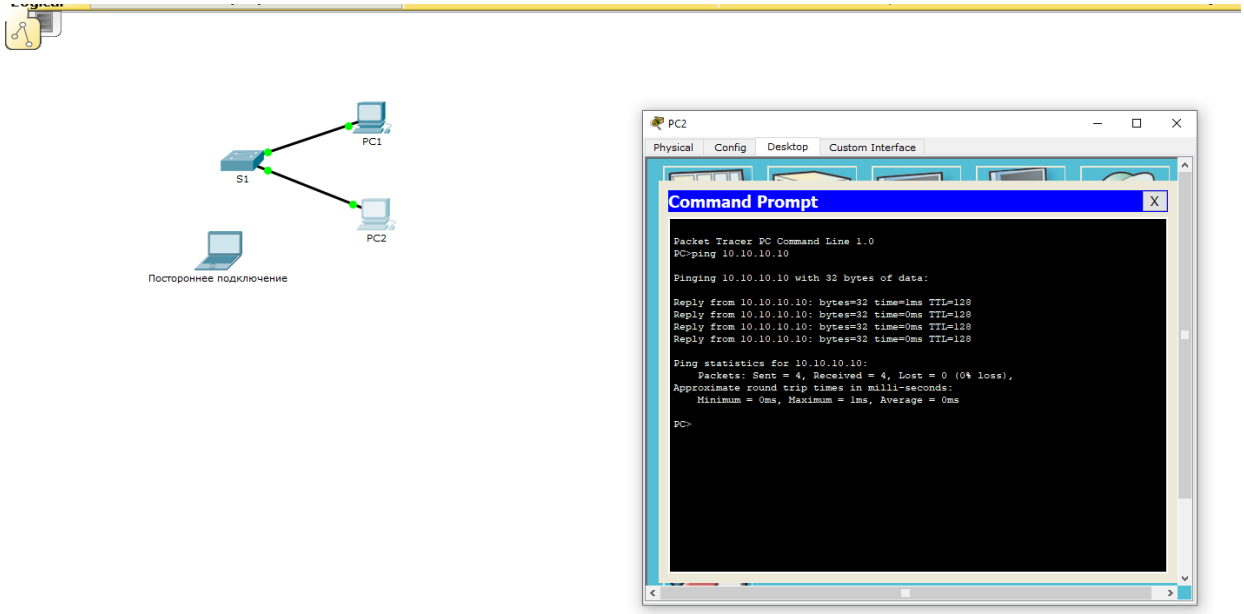

Рисунок 3.23 – Перевірка досяжності комп’ютера PC1 та PC2

Отже, був успішно проведений експеримент, що показує високу ефективність веб-додатку у розв’язанні проблеми безпеки мережі. Для експерименту обрали Restrict режим реагування зі sticky MAC-адресами, які добре емулюють роботу справжньої мережі. Не виникло жодних труднощів з копіюванням коду, та перенесенням його до комутатора.

Даний експеримент зайняв близько 35 хвилин, що на 30 хвилин менше ніж це б було зроблено повністю мануально.

Тому очевидно, що веб-орієнтована система спрощує налаштування в декілька разів, а тому має велику ефективність.

ВИСНОВКИ

Проблема ефективного налаштування безпеки комутаторів досить часто постає перед кожним великим підприємством, що має за мету забезпечення безпеки своєї мережі від зловмисників. Адже захист даних є однією з головних характеристик надійності підприємства, а тому є привабливим, як для співробітників підприємства так і для замовників, що дбають про захист своїх конфіденційних даних.

Практична значущість розробленого у ході виконання веб-додатку полягає в тому, що системному адміністратору не потрібно буде виконувати кожного разу однакову роботу з налаштування мережі, адже для цього потрібно буде лише ввести декілька параметрів за якими ми хочемо одержати конфігураційні налаштування. При розв’язанні задачі ефективного налаштування безпеки портів комутаторів, було з’ясовано головні принципи захисту портів комутаторів для мережі та шляхи забезпечення потрібного рівня захисту, в залежності від поставлених умов.

Таким чином, задача була вирішена в повному обсязі та поставлена ціль була досягнута шляхом розроблення та впровадження веб-орієнтованого додатку за допомогою фреймворку Vue Js. Графічний інтерфейс має дозволяти забезпечення відповідного рівня захисту на комутаторах Cisco. Також було реалізовано зручне копіювання згенерованої конфігурації до справжнього мережевого обладнання, і до обладнання в симуляторі Cisco Packet Tracer.

СПИСОК ЛІТЕРАТУРИ

1. CCNA Semester 2 v6. 0 study Materials and Lab [Електронний ресурс] – https://itexamanswers.net/ccna–semester–2–v6–0–study–materials–labs–online–course. html

2. Що таке ACL та як його налаштовувати [Електронний ресурс] – http://ciscotips. ru/acl

3. Что такое DHCP Snooping и как это работает? [Електронний ресурс] – https://community.fs.com/ru/blog/what-is-dhcp-snooping-and-how-it-works.html

4. Настройка Port Security [Електронний ресурс] – http://ciscomaster. ru/content/nastroyka–port–security–na–kommutatorah–cisco

5. Cisco IOS ACLs [Електронний ресурс] – https://www.pvsm.ru/cisco/17305

6. Грайвороновський М.В. Безпека інформаційно-комунікаційних систем: підручник для ВНЗ / М.В. Грайвороновський, О.М. Новіков; М-вопраці та соц. Політики України. Держнаглядохоронпраці України.- К. : ВНV, 2019. -307с.

7. Хилл Б. Полный справочник по Cisco / Брайан Хилл. – Москва : Издательский дом "Вильямс", 2015. – 1079 с.

8. Уолрэнд Дж. Телекомуникационные и компьютерные сети. Вводный курс / Дж. Уолрэнд. – М.: Постмаркет, 2014. – 480 с.

9. Норенков И.П. Телекомуникационные технологии и сети / И.П. Норенков, В.А. Трудоношин. –М.: МГТУ им. Н.Э. Баумана, 2015. – 389 с.

10. Настройка Cisco Port-Security [Електронний ресурс] – https://wiki.merionet.ru/seti/19/nastrojka-cisco-port-security/

11. Любохинец С. Ответ компании Cisco на современные угрозы безопасности / С. Любохинец // VIII-й Международный Security Innovation Forum 2014. – Киев: 27 ноября 2014 г. – 278 с.

12. Пинженин В. Безопасность сети на основе 802.1x и SFlow, "идеальная и недостижимая" / Владислав Пинженин / Сетевые решения. – 2015. – №3. –331 с.

13. Абаева Б. К. Вопросы проектирования сетей IPTV / Б. К. Абаева // T-Comm – Телекоммуникации и Транспорт. – 2017. –№3. – 432 с.

14. Cisco Systems, Inc. Руководство по технологиям объединенных сетей. – 4-е изд.: [Пер. с англ.] – М.Вильямс, 2017. –№4. – 1040 с.

15. Эндрю Уитакер, Стивен Мак-Квери Маршрутизаторы Cisco: Руководство по конфигурации.- СПб.: Питер, 2016. –№3. – 567 с.

16. Новиков Ю.В., Кондртаенко С.В. Локальные сети: архитектура, алгоритмы, проектирование. – М.: Эком, 2015. –№2. – 322 с.

ДОДАТОК

Додаток А

```
<template>
<template>
  <v-app id="app">
    <div class="top-section">
      <div class="form component">
        <div class="flex flex-center">
          <h1>Port-security</h1>
          <div v-if="errors" class="errors">Заповніть всі поля</div>
        </div>
        <div class="inputs">
          <div class="row">
            <v-text-field
              v-model="port"
              :rules="rules"
              max=24
              label="Порт"
              placeholder="1 - 24"
              outlined
              type="number"
            ></v-text-field>
            <v-text-field
              v-model="max"
              :rules="rules"
              max=24
              disabled
              label="Кількість MAC-адрес"
              placeholder="1"
              type="number"
              outlined
            ></v-text-field>
          </div>
          <div class="row">
            <v-text-field
              v-model="mac"
              :rules="rulesMac"
              label="MAC-адреса"
              @input="macSwitch"
              placeholder="XXXX.XXXX.XXXX"
              outlined
```

```
            ></v-text-field>

            <v-text-field
              v-model="disable"
              label="Порти на відключення"
              placeholder="1 - 24"
              max=24
              outlined
            ></v-text-field>
          </div>
          <div class="row">
            <v-text-field
              v-model="status"
              :rules="rules"
              label="Стан порту"
              placeholder="1 - 24"
              max=24
              outlined
            ></v-text-field>

            <v-select
              :items="items"
              v-model="itemsValue"
              filled
              label="Режим реагування на небезпеку"
              placeholder=" "
          ></v-select>
          </div>
          <div class="flex">
            <div class="row">
              <v-switch  v-model="sticky"  @change="mac  =  ''"  class="ma-2"
label="sticky MAC- адреса"></v-switch>
            </div>
          </div>
        </div>

        <div class="btn-section">
          <v-btn
            color="success"
            class="mr-4"
            @click="showInfoFun"
          >
```

```
                Generate
              </v-btn>

              <v-btn
                color="warning"
              >
                Copy
              </v-btn>
            </div>
          </div>
          <div class="img-component component">
            <img     src="https://kz.all.biz/img/kz/service_catalog/227902.jpeg"
alt="">
          </div>
        </div>
        <div>
          <!-- <img src="@/assets/console.png" alt=""> -->
        </div>
        <div class="component body-section">
          <img src="@/assets/console.png" alt="">
          <div class="text" v-if="showInfo">
          <div>
            en
            <br>
            conf t
            <br>
            int  fa 0/<span>{{port}}</span>
            <br>
            switchport mode access
            <br>
            switchport port-security
            <br>
            switchport port-security maximum <span>{{max}}</span>
            <br>
            switchport port-security violation <span>{{itemsValue}}</span>
          </div>


          <div v-if="mac">
            switchport port-security mac-address <span>{{mac}}</span>
          </div>
          <div v-else>
```

```
        switchport port-security mac-address <span>{{sticky}}</span>
      </div>

        exit
        <br>
        int range fa0/ <span>{{disable}}</span>
        <br>
        shutdown
        <br>
        sh port-security interface fa 0/ <span>{{status}}</span>
        <br>
      </div>

    </div>
  </v-app>
</template>

<script>

export default {
  name: 'App',

  components: {
  },

 data () {
    return {
      port: '',
      max: 1,
      mac: '',
      disable: '',
      status: '',
      itemsValue: '',
      sticky: 'sticky',
      showInfo: '',
      errors: false,
      description: 'California is a state in the western United States',
      rules: [v => v <= 24 || 'Число не має бути більшим за 24'],
      rulesMac: [v => v.length <= 14 || 'Це поле має містити 14 символів'],
      items: ['Protect', 'Restrict', 'Shutdown'],
    }
  },
```

```
        methods: {
          macSwitch() {
            if (this.mac) {
              this.sticky = false;
            } else {
              this.sticky = "sticky";
            }
          },
          showInfoFun () {
            if (this.port && this.max && this.disable && this.status &&
this.itemsValue) {
              this.showInfo = true;
              this.errors = false;
            } else {
              this.errors = true;
            }
          }
        }

      };
      </script>


      <style lang="scss">

      @import "scss/config.scss";
      @import "scss/reset.scss";

      .flex {
        display: flex;
      }

      .flex-center {
        align-items: center;
        justify-content: space-between;
      }

      #app {
        //background-color: #282c34;
        background-color: #b6e0dc;
        background-size: cover;
        background-repeat: no-repeat;
```

```
  background-position: center;
  min-height: 100vh;
  height: max-content;
}

.errors {
  color: red;
}

.top-section {
  display: flex;
}
.component {
  padding: 20px;
  margin: 20px;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.22);
}

.form {
  width: 40%;
}

.img-component {
  width: 60%;
  height: 545px;
  display: flex;
  justify-content: center;
  img {
    height: 100%;
  }
}

.body-section {
  color: #26ff00;
  background-color: #383636;
  padding: 0;
  position: relative;
  img {
    width: 100%;
  }
```

```
  .text {
    position: absolute;
    top: 123px;
    left: 18px;
    font-size: 12px;
    span {
      color: #fff;
    }
  }
}

.inputs {
  margin: 15px 0;
}
#app {
  .v-input--selection-controls {
    margin-top: -10px;
  }
  .v-text-field.v-text-field--enclosed {
    margin: 10px;
  }
  .v-text-field {
    width: 45%;
  }
}

.btn-section {
  margin: 10px 10px 0 10px;
  padding-top: 10px;
  display: flex;
  justify-content: space-between;
}

.row {
  display: flex;
  padding: 0 10px;
}

.checkbox {
  margin-left: 10px;
}
```

```
</style>
<template>
  <div class="main">
    <TypeCard :key="item.title" :item="item" v-for="item of cardList"/>
  </div>
</template>
<script>
    import TypeCard from '../components/layout/Main/TypeCard'
    import {mapMutations} from 'vuex'
    import httpClientAuth from '@/api/httpClientAuth'

    export default {
      name: "Main",
      components: {
        TypeCard
      },
      data() {
        return {
          cardList: [
            {title: 'Створити комірку', route: '/cell-types'},
            {title: 'Створити стек', route: '/stack-types'},
            {title: 'Створити поштомат', route: '/post-box'},
          ],
        }
      },
      computed: {

      },
      methods: {
        ...mapMutations(['switchModal','setName']),
      },
      mounted() {
        console.dir(this.$route.query.code)

        if (this.$route.query.code) {
          let qs = require('qs')

          //const state = this.generateRandomString()
          const state = '8sEDDJzSnNrDPvFQ5QWMidEM4SI29k1C'
          console.log('pkce_state: ' + state)
```

```
                // Create and store a new PKCE code_verifier (the plaintext random
secret)
                //const code_verifier = this.generateRandomString()
                const                          code_verifier                          =
'60b6ed93e92fc4aae9495b608c6c09b07645d4d784c5c826dcf15f29'
                console.log('pkce_code_verifier: ' + code_verifier)

                // Hash and base64-urlencode the secret to use as the challenge
                //const                         code_challenge                         =
this.pkceChallengeFromVerifier(code_verifier)
                const                         code_challenge                         =
'PTZamwyhEivdNcK6YY41sSx6sYwtwxy8SNs8vIDvuN4'
                console.log('code_challenge: ' + code_challenge)


                httpClientAuth.post('http://hydra.sb.np.ua:9001/oauth2/token',
                  qs.stringify({
                    grant_type: 'authorization_code',
                    client_id:      '6adc13a74af267da6f8fdc64503541be.postmachine-
stage.apps.novaposhta.ua',
                    client_secret: 'd5bc0374283b2db6210f2e7a2faafda6',
                    redirect_uri: 'http://pst.wis14.np.ua',
                    code_verifier: code_verifier,
                    code: this.$route.query.code,
                  })
                ).then(response => {
                  debugger
                })
              }
            }
          }
      </script>

      <style scoped lang="scss">

        .main {
          display: flex;
          padding-left: 22px;
          padding-top: 24px;
        }

      </style>
```

```
$text-color: #333333;
$text-color-error: #b00020;
$title-color: rgba(0,0,0, .87);
$subtitle-color: rgba(0,0,0, .6);
$main-bg-color: #e0e0e0;
$focused-color: #3d77d2;
$hover-border-color:  rgba(0, 0, 0, 0.68);

/*--header--*/
$header-line-height: 1.35em;
$header-border-radius: 4px;
$header-border-color: rgba(0, 0, 0, 0.12);
$header-border-color-focused: rgba(61, 119, 211, 1);
$header-shadow-focused: 0 1px 3px rgba(0, 0, 0, 0.15);
$header-color-hot-key: rgba(207, 178, 41, 1);
/*---*/
$hot-key-font: Source Sans Pro;
$border-radius: 4px;
$border-radius-for-widgets: 8px;
$focused-shadow-color: #3d77d2;
$notify-invalid-color: #8e3236;
$focused-invalid-color: #b00020;
$border-color-focused: rgba(61, 119, 211, 1);
/*--*/
$radio-button-enabled-color: #6f6f6f;
$radio-button-enabled-label-color: #3d77d2;

$dialog-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);

/*--table--*/
$border-radius-tr: 2px;
$border-active: #2D95D0;
<template>
    <div class="form-modal">
        <div class="modal-right" v-bind:style="{ width: modalInfo.modalWidth
+ 'px' }">
            <div class="preloader" v-if="loading">
                <v-progress-circular
                :size="50"
                color="primary"
                indeterminate
```

```
                    ></v-progress-circular>
                </div>
                <form @submit.prevent="save"
                      class="form-wrap">
                    <div class="form-content" >
                        <div class="form-content-wrap">
                            <v-btn           text           class="icon-img-btn"
@click="closeModal">
                                <v-icon >arrow_back</v-icon>
                                закрити
                            </v-btn>
                            <div                                    class="form-
title">{{modalInfo.modalTitle}}</div>
                            <div class="main-input-field">
                                <MainInput :inputInfo="mainInputInfo"/>
                            </div>
                            <div class="border-fields-wrap">
                                <BorderField :inputInfo="borderFieldInfo[0]"/>
                                <BorderField :inputInfo="borderFieldInfo[1]"/>
                                <BorderField :inputInfo="borderFieldInfo[2]"/>
                                <BorderField :inputInfo="borderFieldInfo[3]"/>
                            </div>
                        </div>
                    </div>
                    <div class="modal-footer">
                        <button  type="submit"  class="big-btn"  v-bind:class="{
'big-btn-active': vallidForm }">
                            Зберегти
                            <v-icon class="big-btn-icon">play_arrow</v-icon>
                        </button>
                    </div>
                </form>
            </div>
        </div>
    </template>
       computed: {
              ...mapGetters({
                  loading: 'grid/cellTypes/loading',
              }),
            },

            methods: {
```

```
            ...mapActions({
                addItem: 'grid/cellTypes/addItem',
            }),

            closeModal: function() {
                this.$emit('closeModalListener')
            },

            save(e) {
                const formData = new FormData(e.target)
                let payload = {}

                for (const [inputName, value] of formData) {
                    payload[inputName] = value
                }

                this.addItem(payload)
            }
        }
    }
</script>

<style scoped lang="scss">

    @import "../../scss/form-modal.scss";

</style>
```