МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота бакалавра

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ СПИСКІВ КОНТРОЛЮ ДОСТУПУ НА ОБЛАДНАННІ CISCO

Здобувач освіти гр. ІН – 82 Олександр ЧИКАЛОВ

Науковий керівник,
кандидат фізико-математичних наук,
старший викладач Дмитро ВЕЛИКОДНИЙ

Завідувач кафедри
доктор технічних наук, професор Анатолій ДОВБИШ

СУМИ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую_____
Зав. кафедрою Довбиш А.С.
"_____" _____2022 р.

ЗАВДАННЯ
до кваліфікаційної роботи

здобувача вищої освіти четвертого курсу, групи ІН-82 спеціальності «122 – Комп'ютерні науки» денної форми навчання Чикалова Олександра Олександровича.

Тема: «ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ СПИСКІВ КОНТРОЛЮ ДОСТУПУ Н ОБЛАДНАННІ CISCO»

Затверджена наказом по СумДУ
№_____ від _____2022 р.

Зміст пояснювальної записки: 1) літературний огляд за обраною тематикою роботи; 2) постановка завдання для розробки; 3) вибір оптимальних інструментів для розробки мобільного додатку; 4) практична реалізація.

Дата видачі завдання «_____»_____ 2022 р.
Керівник роботи _____Дмитро ВЕЛИКОДНИЙ
Завдання прийняв до виконання _____Олександр ЧИКАЛОВ

РЕФЕРАТ

Записка: 47 стор., 36.рис., 1 додаток, 12 джерел

Об’єкт дослідження - процес розробки програмного забезпечення інформаційної-комунікативної системи автоматичної генерації ACL.

Мета роботи - розробка системи автоматичної генерації ACL.

Методи дослідження – методи обробки інформації, методи розробки програмного забезпечення.

Результат – Розроблений веб-додаток для автоматичної генерації ACL.

АВТОМАТИЗАЦІЯ, ВЕБ-ДОДАТОК, CISCO PACKET TRACER, HTML, CSS, JAVASCRIPT

ЗМІСТ

ВСТУП ……5

1.ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....6

1.1 Визначення списків контролю доступу.....6

1.2 Типи ACL 7

1.3 Функції та структура ACL.....8

1.4 Види ACL 12

1.5 Постановка задачі 14

2.МОДЕЛЮВАННЯ МЕРЕЖІ ЗА ДОПОМОГОЮ ЕМУЛЯТОРА CISCO PACKET TRACER 15

2.1 Емулятор Cisco Packet Tracer 15

2.2 Створення нової мережі на основі емулятора Packet Tracer.....16

2.3 Конфігурація ACL в емуляторі Cisco Packet Tracer 17

3.ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ СПИСКІВ КОНТРОЛЮ ДОСТУПУ 26

3.1 Розробка веб-додатку автоматичної генерації списків на мові програмування JavaScript і технологій HTML і CSS.....26

3.2 Функціональний огляд розробленого веб-додатку.....27

3.3 Тестування веб-додатку автоматичної генерації списків в Cisco Packet Tracer ……33

ВИСНОВКИ.....41

СПИСОК ЛІТЕРАТУРИ.....42

ДОДАТОК А 44

ВСТУП

Сучасні технології розвиваються з неймовірною швидкістю. За 50 років вони розвинулися до таких масштабів, що складно уявити які технології будуть ще через 50 років, але потреба в технологіях захисту та оптимізації буде тільки зростати

Кожна айті компанія велика чи маленька має в своєму розпорядженні багато працівників з різним рівнем кваліфікації. Щоб захисти мережу компанії від втручання некваліфікованого працівника, використовуються списки контролю доступу.

ACL можна використовувати для обмеження потоку даних у мережі та підвищити її продуктивності. Зокрема, списки можна використовувати для того, щоб зменшити навантаження на мережу чи вузол, можна задати різний пріоритет пакетам тим сам пришвидшити роботу мережі.

Оскільки списки доступу налаштовуються вручну, будь-які зміни потребують участі адміністратора для редагування. Великі мережі постійно змінюються, в них постійно додаються або видаляються нові користувачі, кожен користувач повинен мати той чи інший рівень доступу, це створює велике навантаження на адміністратора.

Щоб зменшити навантаження на адміністратора, потрібно проаналізувати роботу списків контролю доступу і створити систему яка буде сама їх налаштовувати.

1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Визначення списків контролю доступу

Список контролю доступу (Access Control List або ACL) – це список прав доступу, що дозволяють або забороняють проводити дії одному суб'єкту над іншим суб'єктом. Зазвичай ACL дозволяє або блокує IP-пакети, але крім цього він може перевіряти, що знаходиться в середині IP-пакета, його тип, TCP і UDP порти. Такий тип фільтрації потрібен в тих ситуаціях коли ми маємо обладнання яке знаходиться на кордоні між інтернетом і приватною мережею. ACL розміщують на вхідному напрямку, що дає змогу блокувати надлишковий трафік. ACL – складається з ряду команд IOS або Internetwork Operating System, які надають змогу визначити, чи переадресовує маршрутизатор пакети або ігнорує їх.[2]

Щоб зрозуміти в чому необхідність використання списків контролю доступу, треба проаналізувати, як відбувається процес передачі даних у мережі. Як нам відомо мережевий трафік надходить у вигляді пакетів. У кожному пакеті міститься певну кількість даних і інформацію необхідну для його доставки. Коли маршрутизатор отримує пакет на будь-який з інтерфейсів, він виконує певний алгоритм дій:

1. Відкриває пакет.
2. Перевіряє адресу призначення отриману з пакету.
3. Шукає адресу призначення в таблиці маршрутизації.
4. Якщо адреси співпадають, то пакет буде перенаправлений з відповідного інтерфейса.
5. У випадку якщо відповідність не знайдена, пакет буде проігнорований.

Такий варіант не підходить більшій частині компаній тому, що будь-який користувач який знає правильну адресу має змогу відправити свій пакет через маршрутизатор. Тому, щоб захисти мережу від втручання, адміністратор мережі може встановити контроль доступу на локальних маршрутизаторах. Як тільки

маршрутизатор буде налаштований, жодна спроба проникнути в захищені ресурси буде відхилена.

1.2 Типи ACL

На даний момент існує два типи ACL - списків: стандартні та розширені. Стандартні ACL шукають відповідність лише IP-адресі відправника, тоді як розширені списки ACL відповідають багатьом полям заголовка пакета. Розширені списки доступу відрізняються від стандартних списків доступу великою різноманітністю полів заголовків пакетів, які використовуються для ідентифікації [12]. Оператор розширеного списку ACL може вказати, що перевіряються кілька елементів заголовка пакета, вимагаючи, щоб усі параметри точно відповідали правилам цього ACL. Ця потужна логіка розпізнавання робить розширені списки контролю доступу більш корисними та складними, ніж стандартні списки керування доступом. Також списки ACL IP бувають нумерованими або іменованими, у конфігурації яких використовуються або номери, або імена. На рисунку 1.1 наведено загальні концепції, пов'язані з категоріями списків ACL.

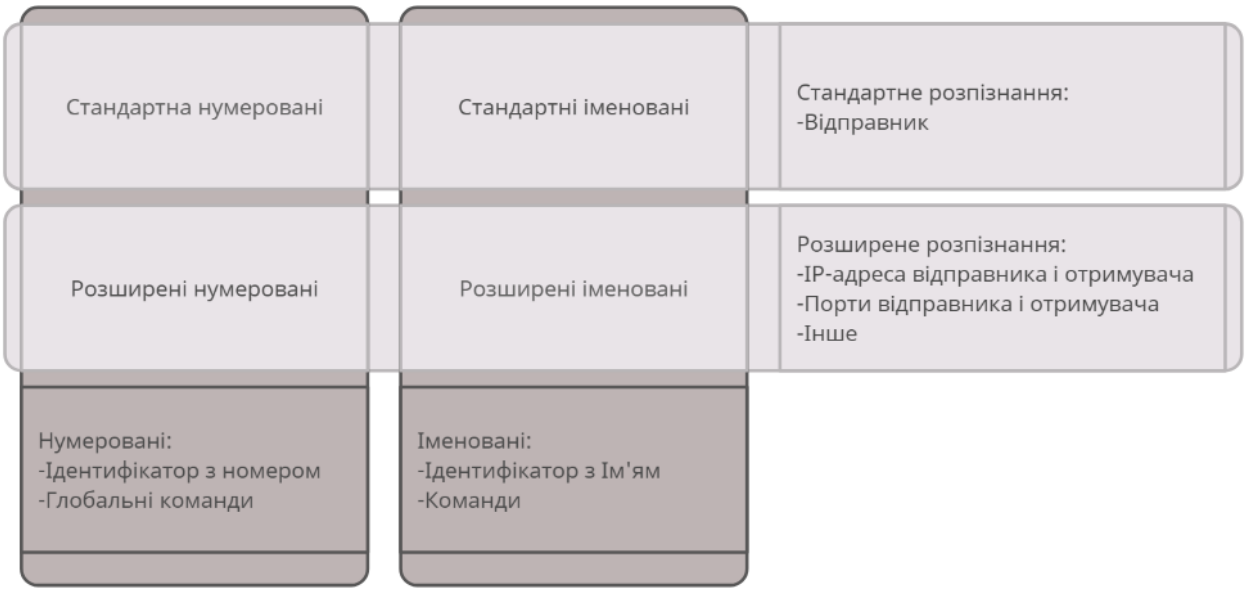

Рисунок 1.1 – Порівняння типів ACL

Стандартні та розширені списки контролю доступу можуть бути створені шляхом створення чисел або імен для виявлення ACL списку та списку його

правил. Маршрутизатор може працювати з різними типами мережевих протоколів, найпопулярнішими з яких є IPv4, IPv6 або IPX, списки контролю доступу ефективно працюють з будь-яким із них. Використання нумерованих списків ACL є ефективним способом визначення типу ACL у невеликих мережах, де зазвичай використовується лише один тип трафіку. Номер не має інформації про призначення ACL-списку. Тому, починаючи з Cisco IOS Release 11.2, списки контролю доступу Cisco використовують імена, призначені спискам.[4]

1.3 Функції та структура ACL

ACL використовуються для забезпечення надійної передачі даних між пристроями в мережі шляхом виконання таких кроків:

- Захист мережі від різних атак, таких як атаки з використанням пакетів IP, протоколу управління передачею (TCP) або протоколу управління повідомленнями Інтернету (ICMP).
- Наприклад, списки ACL можна використовувати для контролю доступу користувачів корпоративної мережі до зовнішньої мережі, для визначення конкретних мережевих ресурсів, доступних користувачам, і для визначення періоду часу, протягом якого користувачі можуть отримати доступ до мережі.
- Обмежити мережевий трафік і підвищити продуктивність мережі. Наприклад, ACL можна використовувати для обмеження пропускної спроможності для вхідного та вихідного трафіку та застосування правил виставлення рахунків до пропускної спроможності, яку запитує користувач, забезпечуючи ефективне використання ресурсів мережі.

Для класифікації пакетів використовуються правила ACL. Коли правила ACL застосовуються до маршрутизатора, він дозволяє або блокує пакети на

основі цих правил. Тому використання правил ACL значно покращує безпеку мережі.

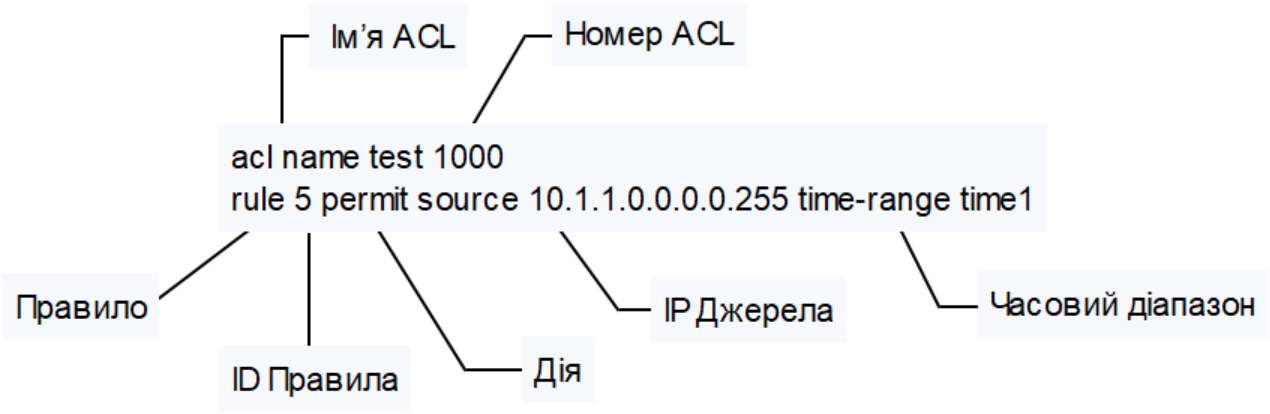

Рисунок 1.2 – Структура ACL

Ім’я ACL: ідентифікує іменований ACL, подібний до представлення доменного імені IP-адреси. Ім'я ACL не можна змінити після його створення. Повторювані імена ACL можна використовувати лише між базовим ACL і базовим ACL, а також між розширеним ACL і розширеним ACL6. Ви можете додати число, визначаючи іменований ACL. Якщо не вказати номер, система автоматично призначає максимальну кількість доступних діапазонів на основі типу ACL імені ACL. [4]

Номер ACL: ідентифікує нумерований ACL (якщо визначено окремо). Для нумерованих списків керування доступом діапазон нумерації залежить від типу списку керування доступом.

Правило: Визначає умови для збігу пакетів. Ідентифікатор правила: визначає правила ACL, які можуть бути налаштовані вручну або автоматично призначені системою.

Ідентифікатори правила ACL варіюються від 0 до 4294967294. Пакети та правила ACL порівнюються в порядку зростання ідентифікатора правила. Пристрій припиняє збіг пакетів, доки не буде знайдено відповідність. [4]

Дія: вказує, як обробляти пакет, включаючи дозвіл і заборону.

Умова відповідності: вказує критерії, яким повинен відповідати пакет, щоб відповідати правилу. ACL підтримує різноманітні умови відповідності. На додатку до вихідної IP-адреси та діапазону часу на Рис 1.2, інші умови відповідності включають інформацію заголовка кадру Ethernet рівня 2 (MAC-адреса джерела, MAC-адреса призначення та тип протоколу Ethernet), інформацію про пакети рівня 3, IP-адресу призначення та тип протоколу та інформація про пакети рівня 4 (номер порту TCP або UDP).[4]

Категорія	Визначення правила	Числовий діапазон
Interface-based ACL	Визначає правила на основі вхідних пакетів інтерфейсів.	1000-1999
Basic ACL	Визначає правила на основі вихідних IP-адрес, інформації про фрагментацію та часові діапазони.	2000-2999
Advanced ACL	Визначає правила на основі вихідних IPv4-адрес, адрес IPv4 призначення, типів протоколів IPv4, типів ICMP, номерів портів джерела/призначення TCP, номерів портів джерела/призначення UDP та діапазонів часу.	3000-3999
Layer 2 ACL	Визначає правила на основі інформації в заголовках кадрів пакетів Ethemet, наприклад, MAC-адреси джерела, MAC-адреси призначення та типи протоколів рівня 2.	4000-4999
User-defined ACL	Визначає правила на основі заголовків пакетів, зміщень, масок рядка, символів і рядків символів, визначених користувачем. ACL виконує операцію I над байтами пакета з певної позиції за заголовком пакета та маскою рядка символів. Потім ACL порівнює витягнутий рядок символів із визначеним користувачем рядком символів.	5000-5999
User ACL	Визначає правила на основі вихідних IPv4-адрес або груп керування списком користувачів (UCL)/адрес IPv4 призначення або груп UCL призначення, типів протоколів IPv4, типів ICMP, номерів портів джерела/призначення TCP та номерів портів джерела/призначення UDP.	6000-6999

Рисунок 1.3 – Класифікація ACL

Відповідно до попереднього процесу встановлення після фільтрації пакетів за правилами ACL можуть бути отримані наступні два результати:

- Пакети відповідають правилам в ACL
- ACL не існує, ACL не містить правил або пакетів, які не відповідають жодному з правил у ACL.

Дозволені або заборонені пакети визначаються правилами і діями ACL, зазначеними в сервісному модулі, до якого застосовується ACL. Різні сервісні

модулі по-різному обробляють пакети, відфільтровані за правилами ACL. Наприклад, модуль Telnet безпосередньо пересилає пакети, які відповідають правилам дозволів, тоді як модуль політики трафіку відхиляє пакети, які відповідають правилам дозволів, якщо відхиляє поведінку, налаштовану в політиці трафіку.

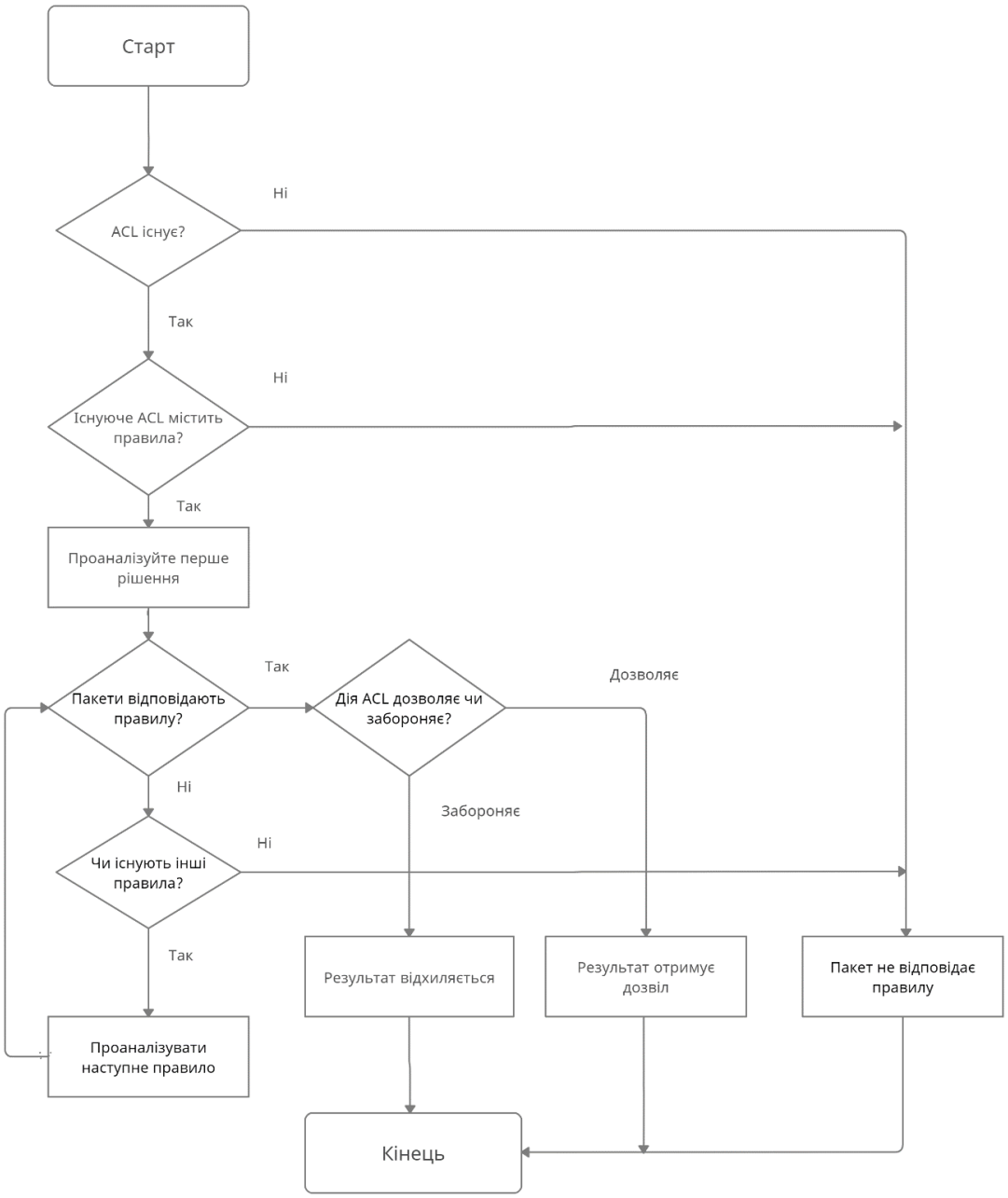

Рисунок 1.4 – Механізм зіставлення ACL

ACL складається з кількох правил, які можуть перетинатися або конфліктувати. Наприклад, ACL містить два правила:

- rule deny ip destination 192.100.1.0 0.0.0.255 - Пакети, призначені для IP-адреси у сегменті мережі 192.100.0.0/16, будуть відхилені.
- rule permit ip destination 192.100.1.0 0.0.0.255 - Дозволяє пакети, призначені для IP-адреси в сегменті мережі 192.100.1.0/24, який менший, ніж сегмент мережі 192.120.0.0/16.

Система спочатку зіставляє пакет, призначений для IP-адреси 192.10.1.1, із правилом заборони пакет відкидається. Однак, якщо система спочатку порівнює пакет із правилом дозволу, пакет пересилається. Тому, якщо правила ACL перекриваються або конфліктують, порядок відображення визначає результат зіставлення.

Порядок налаштування такий, система зіставляє пакети з правилами ACL у порядку зростання значень ідентифікаторів правил. Тобто спочатку проходить перевірка правилом з найменшим ідентифікатором. Якщо для правила вказати менше значення для ідентифікатора правила, воно набирає чинності раніше, ніж правило з більшим значення ідентифікатором. Якщо для правила вручну не вказано значення ідентифікатора, система надає йому своє значення. Цей ідентифікатор правила є найбільшим в ACL і має мінімальне збільшення. Отже, це правило діє в останню чергу.[4]

Система організовує правила ACL відповідно до точності правил ACL за принципом глибини і перевіряє, чи відповідають пакети даних правилам у порядку точності від високої до низької. Це правило визначає найсуворіші умови з найвищою точністю і має найвищий пріоритет.

1.4 Види ACL

Рефлексивні списки контролю доступу (також відомі як інструменти фільтрації IP-сеансів) допомагають запобігти порушенням класу безпеки, дозволяючи кожному дозволеному сеансу TCP або UDP проходити через

пристрій окремо. Для цього маршрутизатор повинен якимось чином реагувати, виявляючи перший пакет нового сеансу зв’язку між двома хостами. У відповідь на пакет маршрутизатор додає оператор permit до списку контролю доступу, дозволяючи трафіку проходити через сеанс, що характеризується використанням певних IP-адрес відправника та одержувача та певних портів TCP або UDP. Традиційні розширені списки контролю доступу дозволяють трафік, дозволяючи двонаправлену пересилку пакетів між будь-якими двома IP-адресами, але вимагають додаткової аутентифікації порту TCP в протоколі HTTP.

Рефлексивні списки керування доступом повною мірою дозволяють законним користувачам передавати та приймати пакети через маршрутизатор, але відкидати всі пакети, що надходять від інших хостів, на кшталт пакетів, відправлених зломщиком. Якщо застосовуються рефлексивні списки керування доступом, то відразу після створення нового сеансу користувачем на підприємстві маршрутизатор виявляє цей новий сеанс і реєструє IP-адреси відправника та одержувача, а також порти, що використовуються у цьому сеансі.

Динамічні списки контролю доступом дозволяють вирішувати різні проблеми, які також потребують великих зусиль при спробі вирішити їх за допомогою традиційних списків керування доступом. Наприклад, потрібно надати доступ до кількох серверів для невеликої групи користувачів. Однак якщо користувач сидить на іншому комп’ютері, або тимчасово отримує нову адресу через DHCP, або забирає свій ноутбук додому тощо, авторизація все одно залишається за користувачем. У зв'язку з цим традиційний список керування доступом повинен бути відредагований для підтримки кожної нової IP-адреси. З часом обслуговування такого списку контролю доступу, яке включає в себе перевірку всіх подібних IP-адрес, стає все більш трудомістким, цю проблему можна вирішити зв’язавши застосування списку контролю доступом з процесом аутентифікації користувача. Але в цьому випадку користувачів необхідно проінструктувати, щоб вони починали свою роботу не зі спроби підключення до сервера, а з встановлення сеансу зв'язку з Telnet з маршрутизатором. [3]

Списки керування доступом, контрольовані за часом, під списками управління доступом, контрольованими за часом, маються на увазі стандартні списки управління доступом IP і нумеровані, і іменовані, які мають одну особливість: вони дозволяють додавати обмеження часу в команди конфігурації. У деяких випадках може знадобитися перевірка пакетів з урахуванням критеріїв у списку керування доступом, але лише у певний час або навіть у певні дні тижня. У списках керування доступом, що контролюються за часом, передбачена можливість додавати тимчасові обмеження, у зв'язку з чим система IOS вводить або видаляє оператор зі списку керування доступом після настання встановленого часу.

1.5 Постановка задачі

Виконавши аналіз великого обсягу літератури, можна сформувати мету даної наукової роботи: необхідно розробити інформаційно комунікаційну систему, яка буде проводити автоматичну генерацію списків контролю доступу. Система повинна забезпечувати просте та зручне перенесення згенерованого коду на реальне обладнання Cisco.

Програмне забезпечення повинно дозволяти навіть не кваліфікованим користувачам встановити списки контролю доступу на будь-якій мережі Ethernet. Для створення інформаційно комунікаційної системи автоматичної генерації списків необхідно розробити систему в якій проводитися автоматичний аналіз користувача з подальшою видачою певних видів дозволу. В результаті система стане автоматично видавати дозволи усім користувачам в залежності від їх ієрархії в системі.

Постановка задачі:

1. Налаштування конфігурації в мережі Cisco.
2. Розробка системи автоматичної генерацій списків.
3. Тестування розробленої системи на обладнанні Cisco.

2. МОДЕЛЮВАННЯ МЕРЕЖІ ЗА ДОПОМОГОЮ ЕМУЛЯТОРА CISCO PACKET TRACER

2.1 Емулятор Cisco Packet Tracer

Cisco Packet Tracer — це потужний симулятор мережі, який дозволяє системним адміністраторам експериментувати взаємодіяти з поведінкою мережі та проводити оцінку можливих сценаріїв. Packet Tracer дозволяє імітувати роботу різних мережевих пристроїв: маршрутизатори, комутатори, бездротові точки доступу, ПК, мережеві принтери, IP-телефони тощо. Використання інтерактивного симулятора дає дуже правдоподібне відчуття роботи з мережею, створюючи мережі на десятки або сотні пристроїв. Ці налаштування, в свою чергу, залежать від характеру пристрою: деякі можна налаштувати за допомогою команд Cisco IOS, графічного веб-інтерфейсу або командного рядка операційної системи або графічні меню. [11]

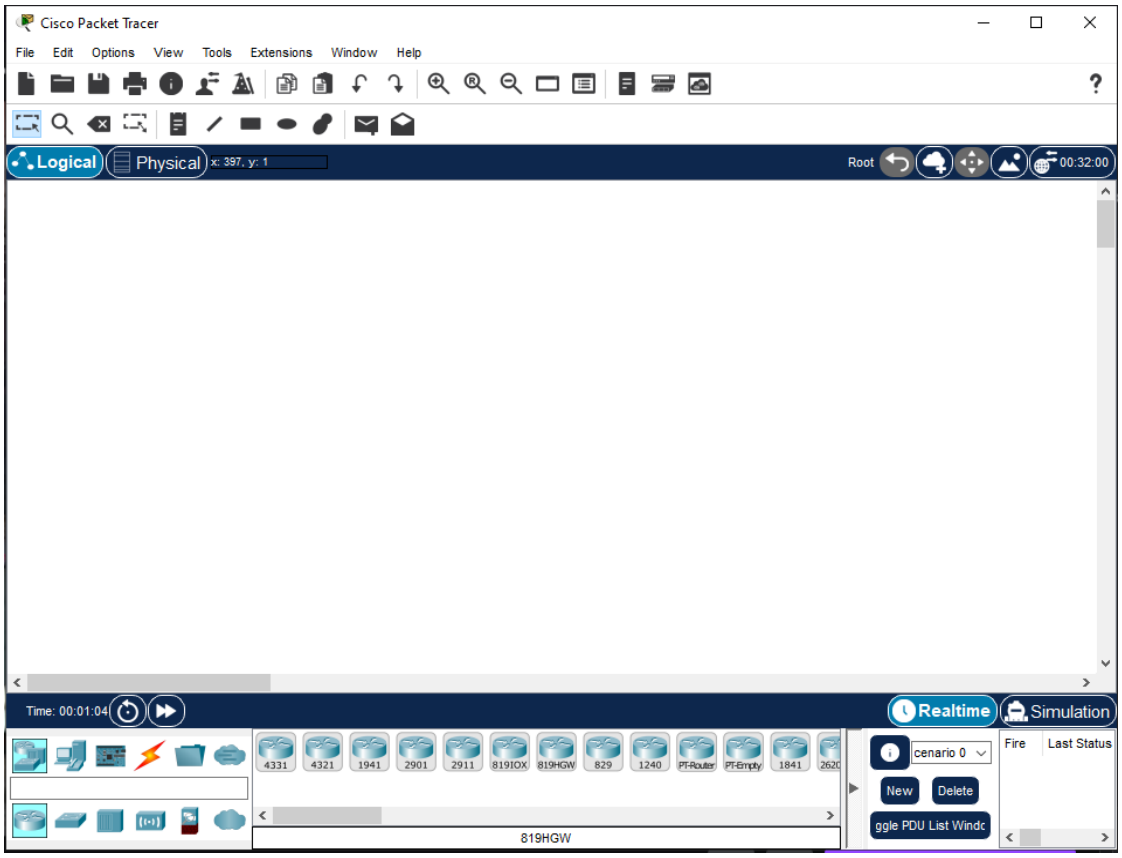

Рисунок 2.1 – Інтерфейс Cisco Packet tracer.

Cisco Packet Tracer можна використовувати не тільки як емулятор, а й як мережевий додаток, що моделює віртуальні мережі в реальних мережах, у тому числі в Інтернеті. Користувачі різних комп'ютерів, незалежно від їхнього розташування, можуть працювати, налаштовувати та усувати неполадки в одній топології мережі. Ця функція багатокористувацького режиму Cisco Packet Tracer широко використовується для командної роботи.

Крім того, за допомогою Cisco Packet Tracer користувачі можуть отримати навички проектування, моделюючи побудову логічних і фізичних моделей мережі. Схема мережі може бути накладена на креслення реального будинку або навіть міста, а всі його кабелі спроектовані, розташовуючи обладнання в будівлях і приміщеннях, враховуючи такі фізичні обмеження, як довжина і тип кабелів або радіус прокладки, зона покриття бездротової мережі. Симуляція, візуалізація, багатокористувацький дизайн і можливості проектування роблять Cisco Packet Tracer унікальним інструментом для вивчення мережевих технологій.

2.2 Створення нової мережі на основі емулятора Packet Tracer

Першим кроком для реалізації автоматичної генерації списків є створення мережі в якій користувачі мають різні посади такі як бухгалтер, адміністратор, айті-спеціаліст, спеціаліст технічної підтримки та інші, кожен з них повинен мати певний рівень доступу, в цьому допоможуть списки контролю доступу.

Було виконано конфігурацію наступного сценарію (Рис. 2.2). Дана модель складається з 6 мереж: розробка, підтримка, тестування, адміністрування, DNS-сервер і серверна частина. Така конфігурація допоможе пояснити взаємозв'язок між різними компонентами системи та налаштувати поведінку мережі, надавши приклади з реального світу.

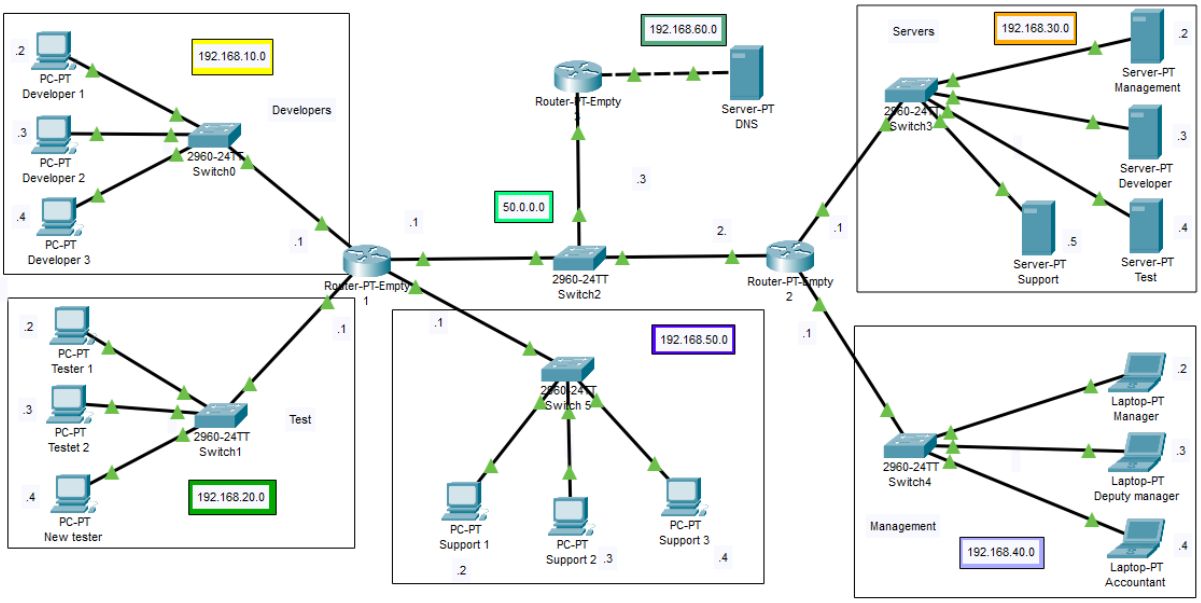

Рисунок 2.2 – Змодельована мережа.

Система складається з шести мереж. Відділ розробки, відділ тестування та відділ підтримки використовують по три комп’ютера, а відділ менеджменту складається з трьох ноутбуків, якими користуються відповідно керівник компанії, системний адміністратор і бухгалтер. . У розділі серверів є 4 сервери, відповідальні за кожен відділ. Доступ до серверів організований за технологією DNS, сервер якого знаходиться в мережі. Кожен користувач і сервер налаштовуються за допомогою IP-адреси. Мережа з'єднана між собою шістьма комутаторами і трьома маршрутизаторами, кожен з яких має статичні маршрути.

2.3 Конфігурація ACL в емуляторі Cisco Packet Tracer

Щоб краще бачити які процеси відбуваються в мережі було прийняте рішення налаштувати розширені ACL, змоделювавши те як буде відбуватися автоматична генерація списків контролю доступу. Було вирішено налаштувати найпоширеніші списки розширеного контролю доступу, які можуть стати вирішенням певних задач.

Першим з найпоширеніших налаштувань розширених списків доступу є заборона трафіку від одного вузла до іншої мережі. Цей тип блокування часто

використовується, коли приймається на роботу новий співробітник і давати доступ до даних компанії ще зарано. На даний момент відділ тестування є новим відділом і має доступ до відділу розробки, це може викликати порушення безпеки, так як кожен новий співробітник може влізти в іншу мережу і нашкодити усій компанії.

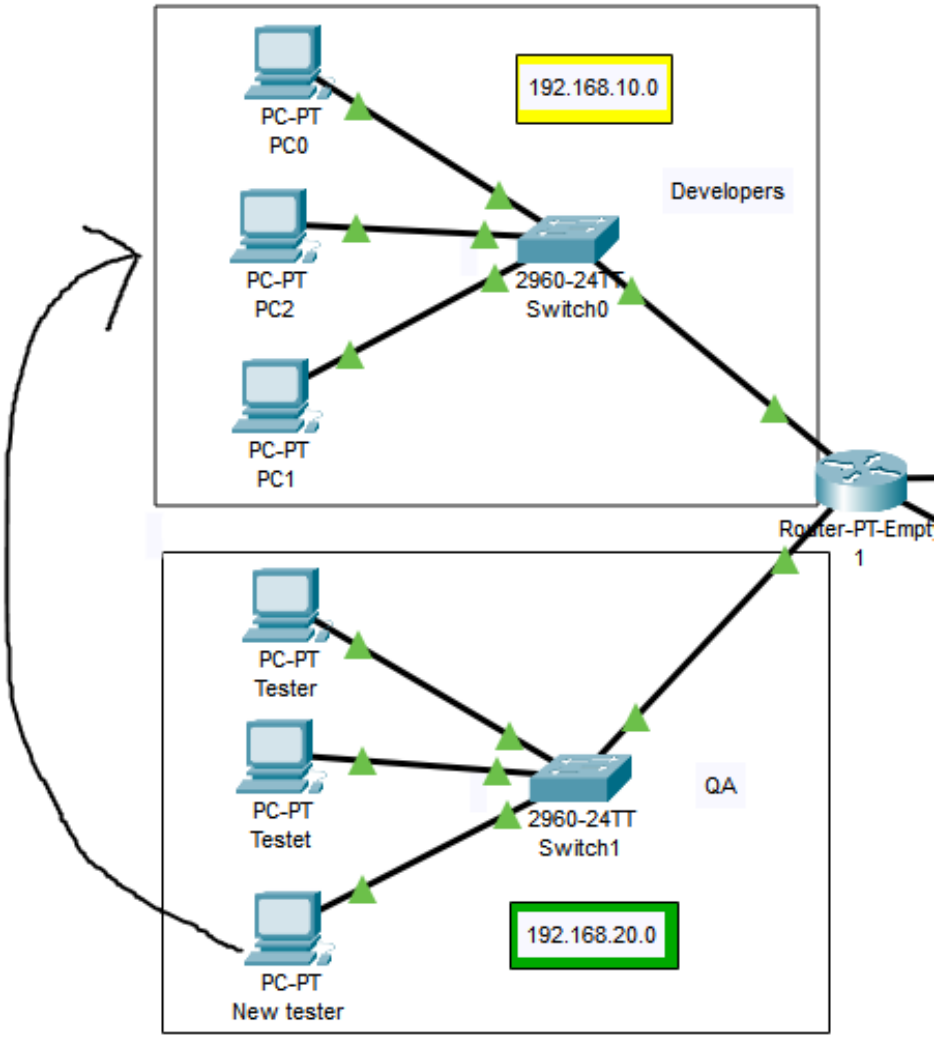

Рисунок 2.3 – Схема взаємодії між New tester і мережею Developers

Тому першим чином потрібно обмежити доступ новим співробітникам відділу тестування до відділу розробників. Щоб обмежити доступ потрібно виконати наступний код на маршрутизаторі 1.

```
enable
configure terminal
```

```
access-list 134 deny icmp host 192.168.20.4 192.168.10.0 0.0.0.255
int fa1/0
ip access-group 134 in
ex
```

Тепер список контролю доступу під номером 134 забороняє вузлу 192.168.20.4 спрямовувати свій трафік до мережі 192.168.10.0. Щоб продемонструвати це використовуємо команду ping і бачимо, що трафік заблоковано (Рис 2.4).

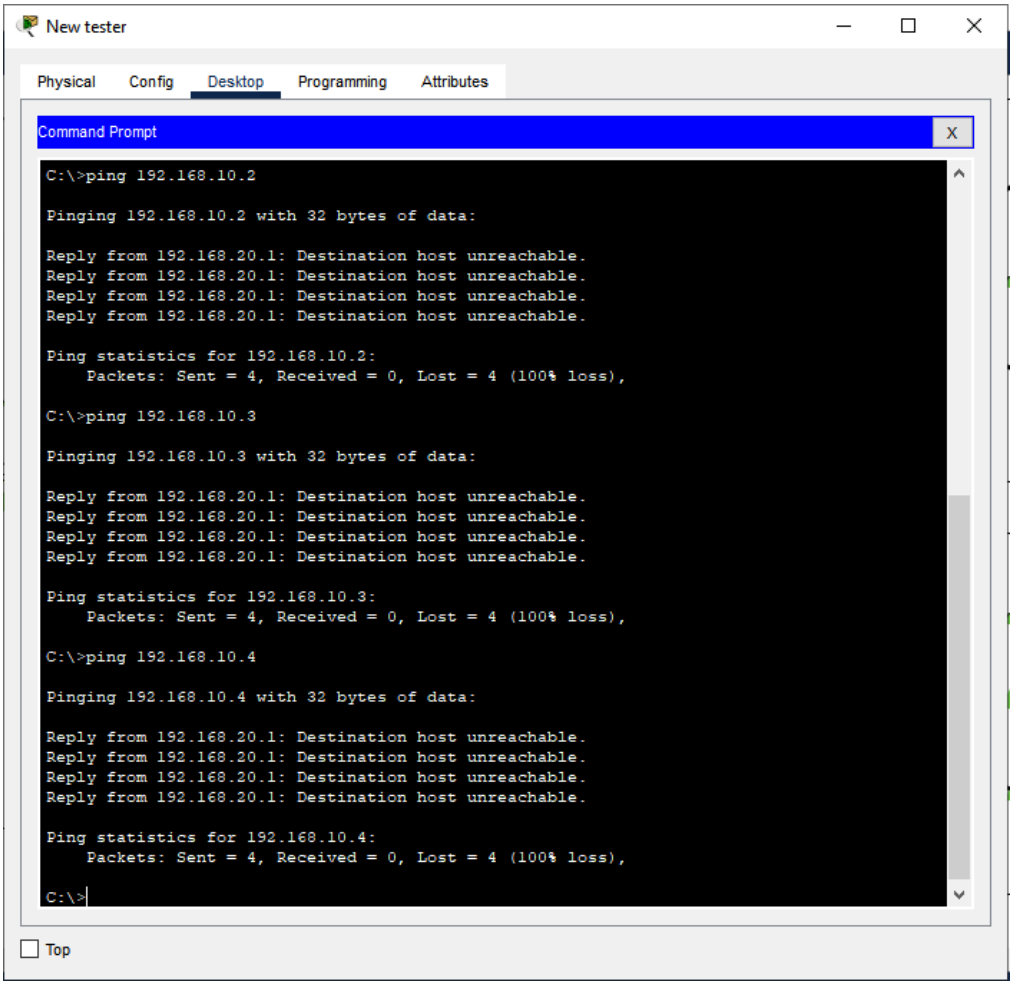

Рисунок 2.4 – Пінг з вузла New tester у мережі тестувальників та розробників

Наступним кроком буде обмеження трафіку від мережі до вузла. Мережа тестувальників не повинна мати доступ до серверів розробки, менеджменту та технічної підтримки.

```
enable
configure terminal
access-list 127 deny icmp 192.168.20.0 0.0.0.255 192.168.30.2
access-list 127 deny icmp 192.168.20.0 0.0.0.255 192.168.30.3
access-list 127 deny icmp 192.168.20.0 0.0.0.255 192.168.30.5
int fa2/0
ip access-group 127 in
no sh
 ex
```

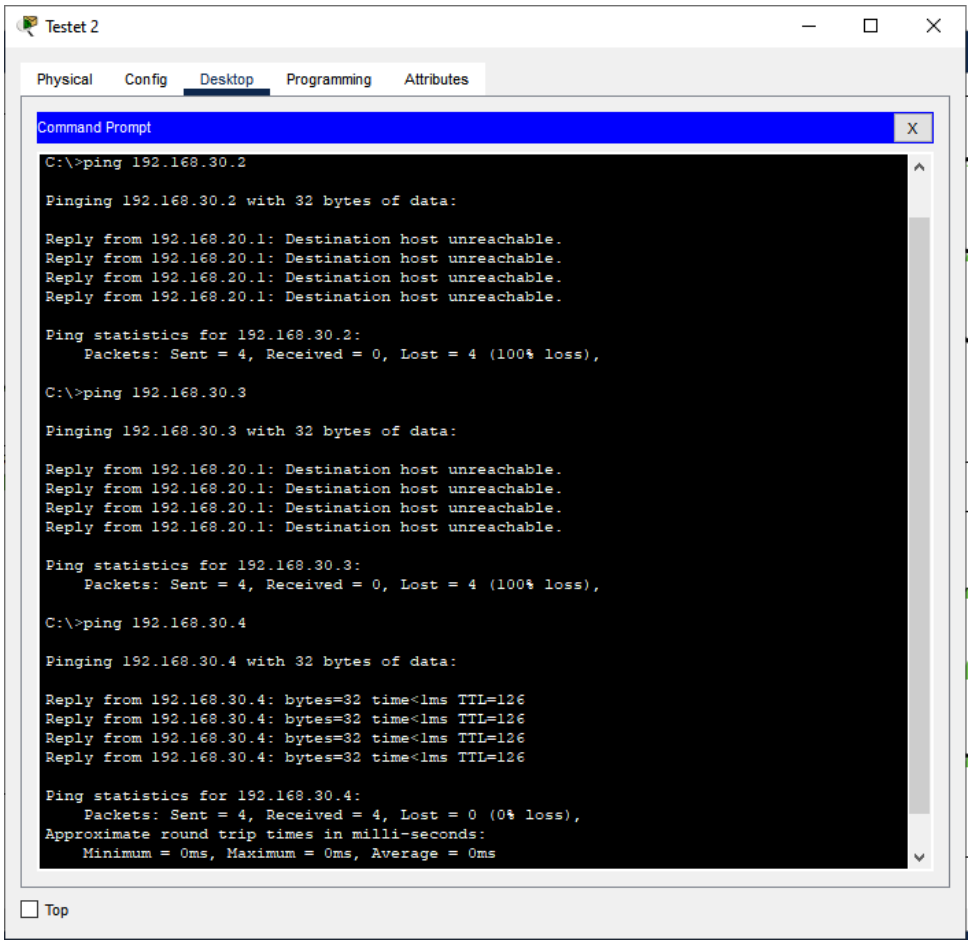

Рисунок 2.5 – Пінг з вузла New tester у мережу тестувальників і серверів.

Після налаштування списків контролю доступу для серверів потрібно також виконати налаштування між мережевого захисту. Такий тип захисту є представником блокування від мережі до мережі. Так як в будь якій компанії

штат працівників постійно змінюється і щоб запобігти порушення безпеки треба обмежити доступ працівникам із відділів тестування і підтримки.

enable

configure terminal

access-list 156 deny icmp 192.168.20.0 0.0.0.255 192.168.40.0 0.0.0.255

int fa2/0

ip access-group 156 in

no sh

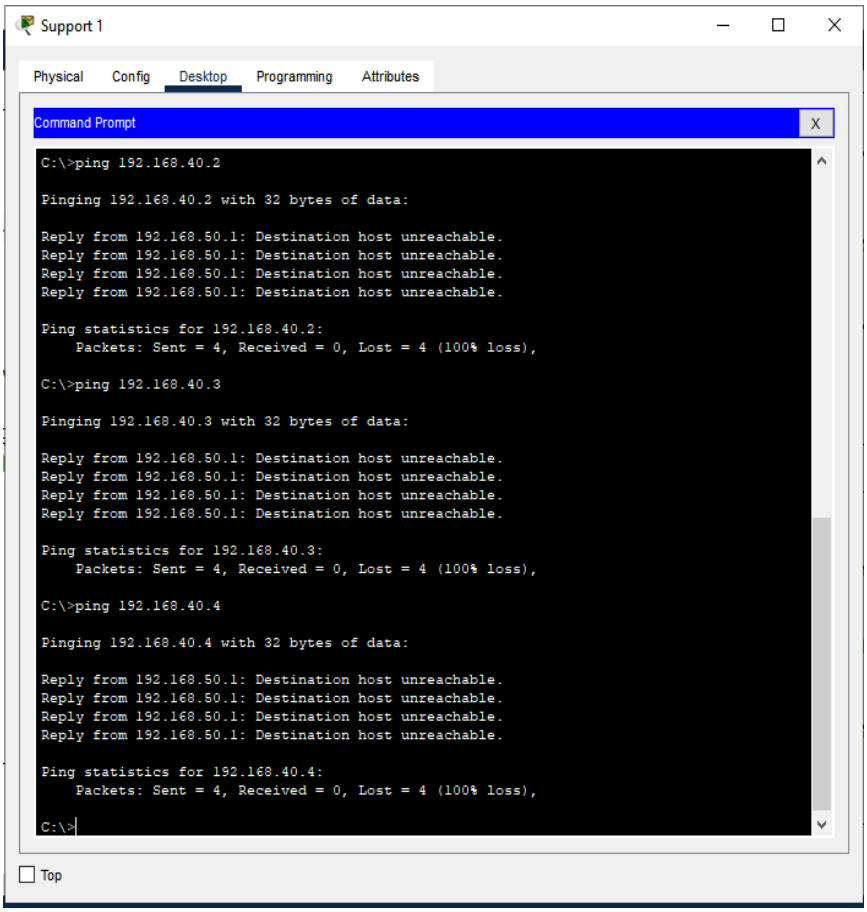

Рисунок 2.6 – Пінг з мережі Support до мережі менеджменту неможливий

enable

configure terminal

access-list 148 deny icmp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255

int fa3/0

ip access-group 148 in

no sh

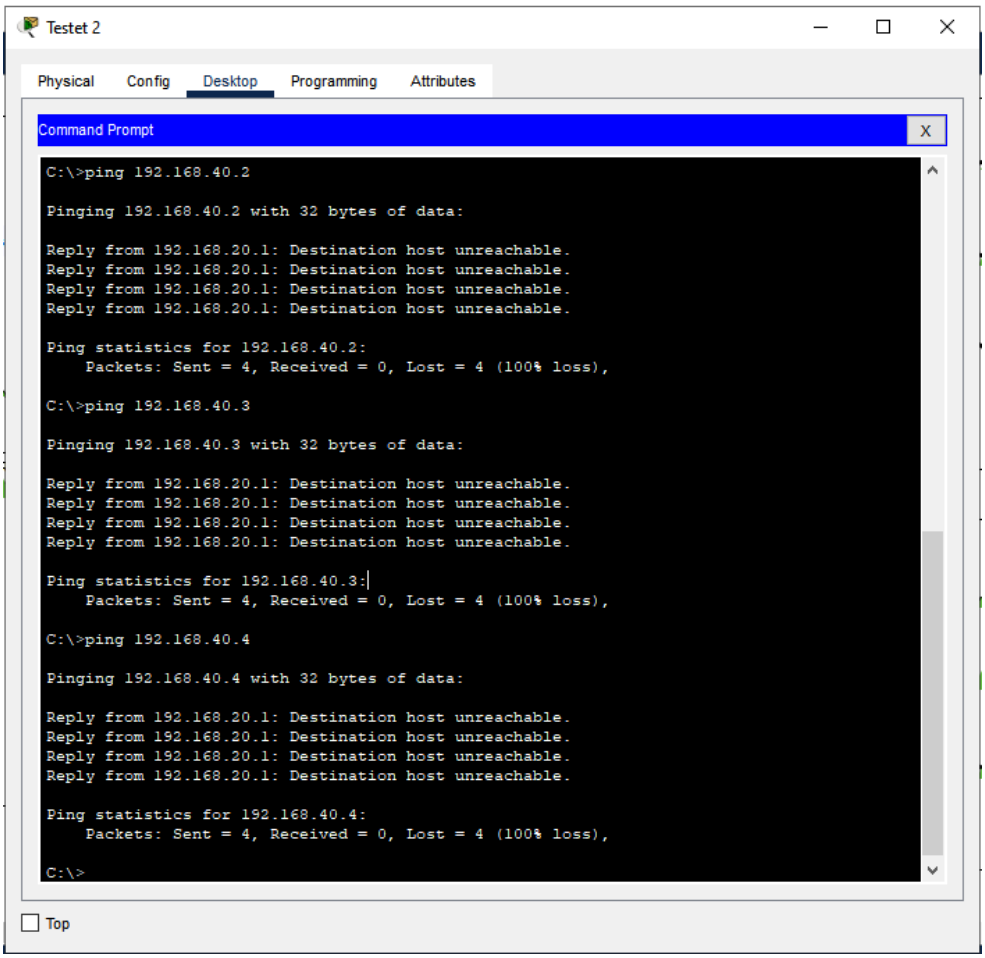

Рисунок 2.7 – Пінг з мережі Test до мережі менеджменту неможливий

Щоб захисти сервери від перевантаження, необхідно відключити можливість надсилання запитів ping на них, але потрібно залишити доступ співробітникам. Даний тип захисту допоможе захистити сервери від різних помилок в системі які можуть нашкодити роботі мережі, наприклад, перевантажити сервер великою кількістю запитів.

Щоб захистити сервери, потрібно налаштувавши наступний список контролю доступу на маршрутизаторі 2.

access-list 193 deny icmp any any echo

int fa1/0

ip access-group 193 out

no sh

ex

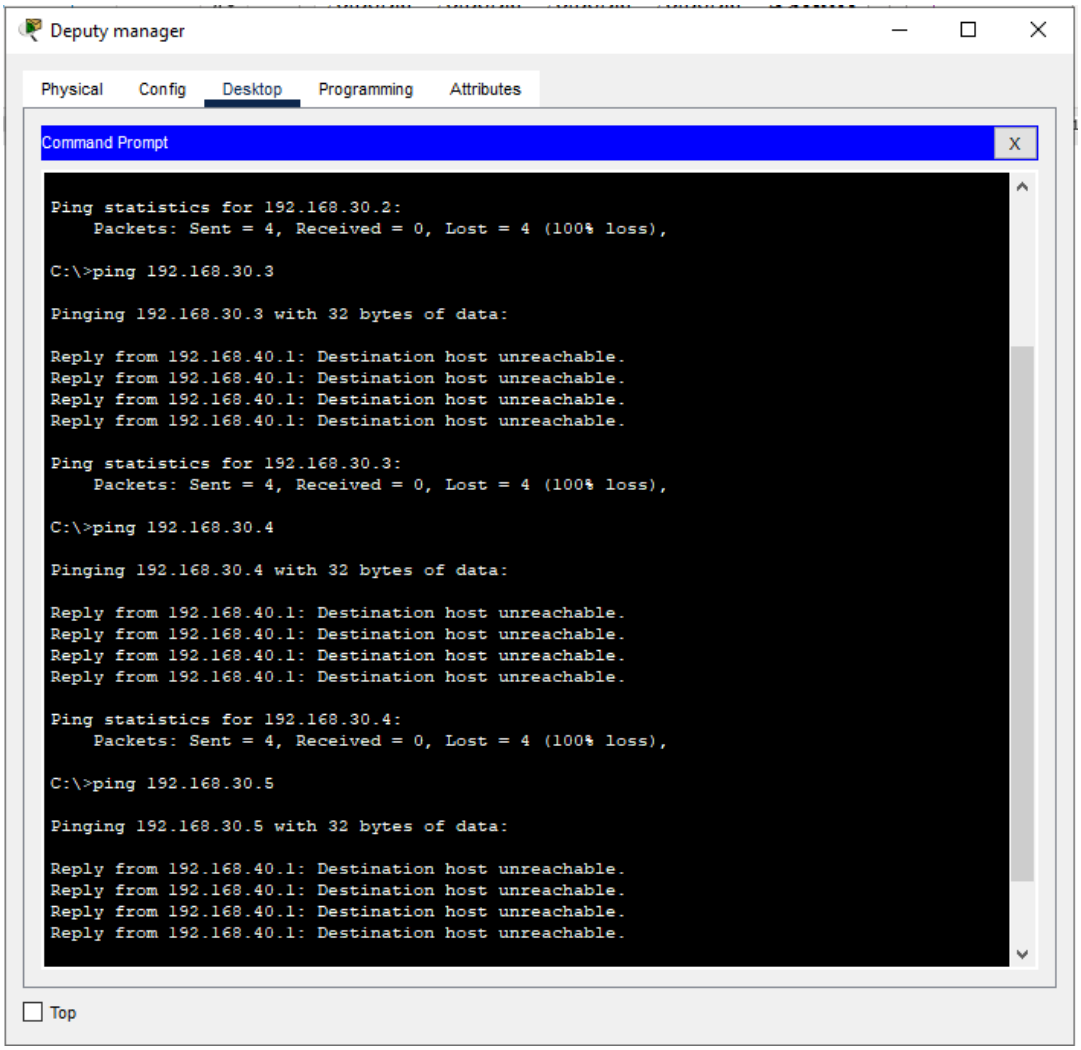

Рисунок 2.8 – Пінг з вузла Deputy manager до мережі Seprvers

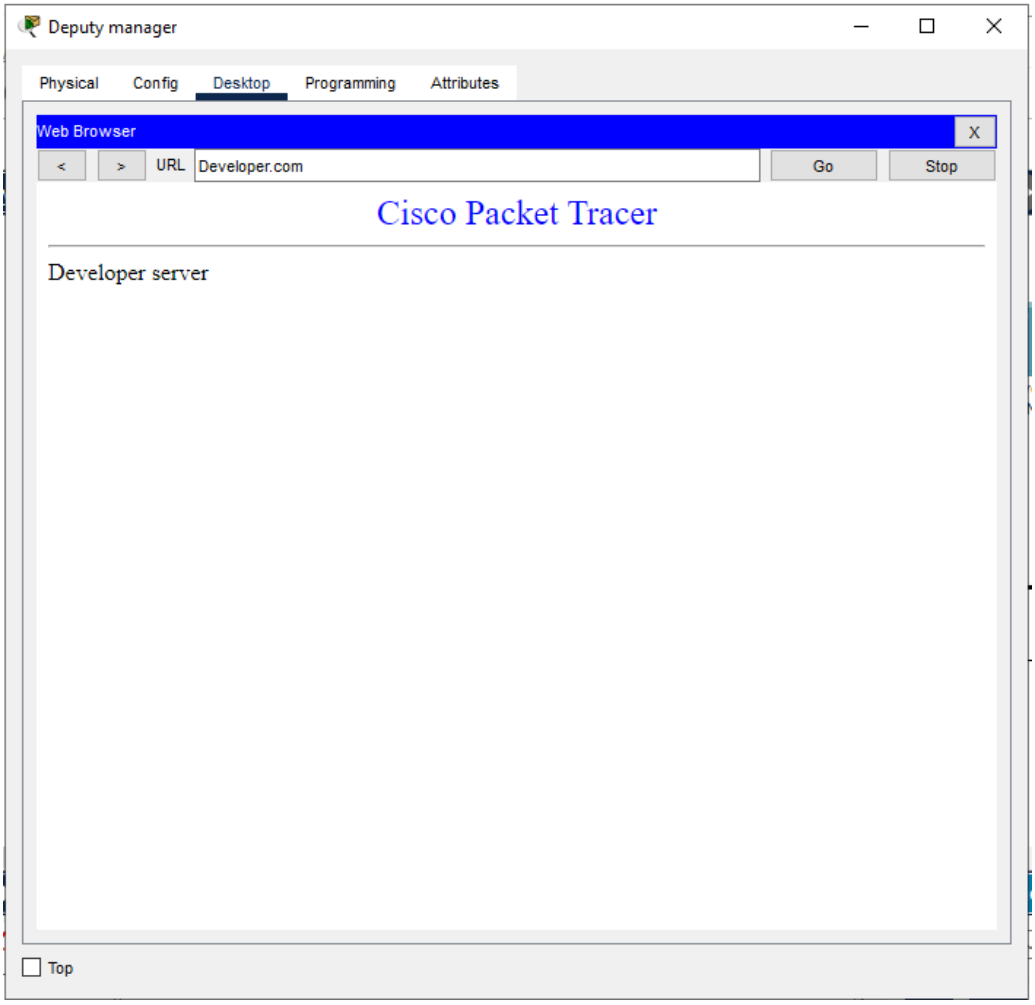

Рисунок 2.9 – перевірка доступу до сервера Developer з вузла Deputy manager

Тепер потрібно заборонити трафік від вузла один до вузла два. Такий спосіб захисту потрібен для нових працівників, так як не кваліфікований спеціаліст може дуже нашкодити системі. Тому робітнику New tester буде дозволено мати трафік з робітником Developer 1. Потрібно відредагувати список контролю доступу під номером 134.

enable

configure terminal

ip access-list

ip access-list e 134

(config-ext-nacl)#no 10

(config-ext-nacl)#no 20

```
access-list 134 deny icmp host 192.168.20.4 192.168.10.3 0.0.0.0
access-list 134 deny icmp host 192.168.20.4 192.168.10.4 0.0.0.0
int fa2/0
ip access-group 134 in
no sh
ex
```

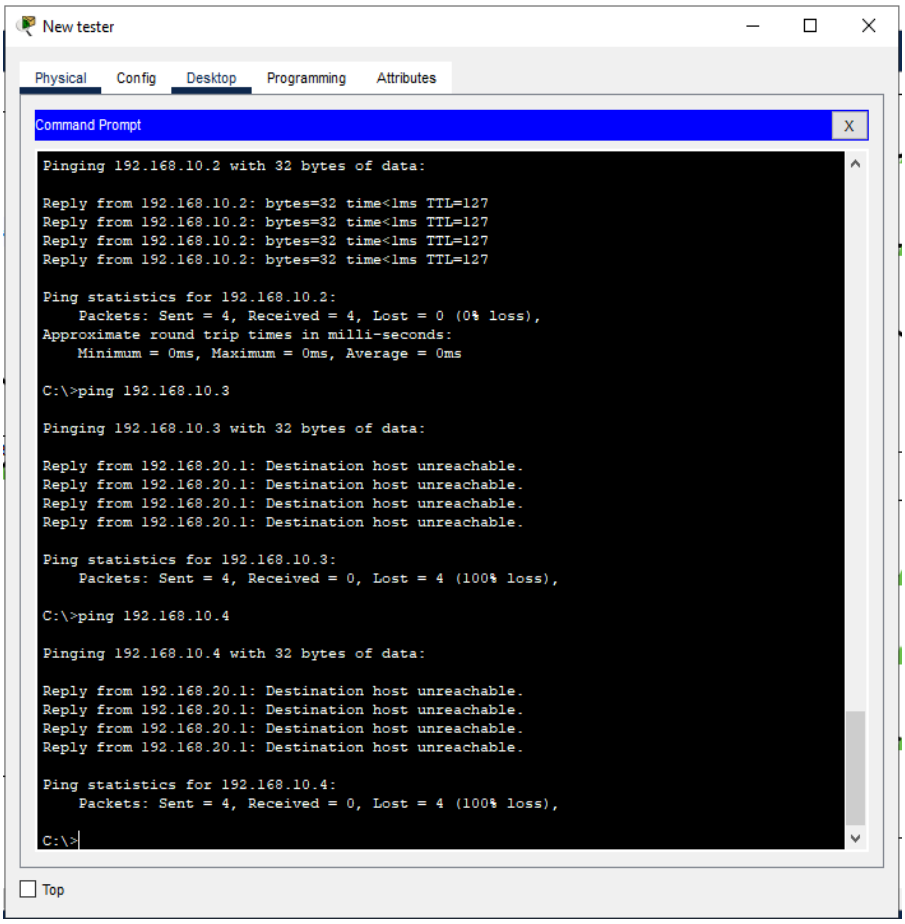

Рисунок 2.10 – Перевірка доступу вузла New tester до вузлів Developer

Всі зроблені налаштування можуть виглядати складними для не кваліфікованого користувача, але єдина їх проблема це час, вони вимагають багато часу та знання конфігурації мережі. Отже необхідно розробити систему, яка буде автоматично визначати ким є кожний користувач мережі і відповідно до нього генерувати списки контролю.

3.ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ СПИСКІВ КОНТРОЛЮ ДОСТУПУ

3.1 Розробка веб-додатку автоматичної генерації списків на мові програмування JavaScript і технологій HTML і CSS

Щоб почати розробку веб-додатку, потрібно визначитися, що потрібно для його створення. Вибір веб-додатку обумовлений тим, що в ньому з легкістю створюється інтерфейс, яким зможе користуватися кожна людина в незалежності від її рівня обізнаності в даній сфері.

HyperText Markup Language або HTML стандартизована мова гіпертекстової розмітки документів для перегляду веб-сторінок у браузері. Використовуючи HTML, можна вставляти різні дизайни, зображення та інші об’єкти (наприклад, інтерактивні веб-форми) у відображені сторінки.

Абревіатура CSS розшифровується як Cascading Style Sheets, що означає «каскадні таблиці стилів». це мова розмітки для візуального дизайну веб-сайту.

Основна мета CSS полягає в тому, щоб відокремити опис логічної структури веб-сторінки створеної за допомогою HTML або іншої мови розмітки від опису зовнішнього вигляду. Відтворення забезпечує більшу гнучкість та керованість , а також зниження складності та повторюваності структурованого контенту.

Крім того, CSS дозволяє відтворювати один і той же документ з різними стилями або методами виведення.. Це значно спрощує роботу і скорочує витрати часу. Один створений файл стилю може бути розподілений на багатьох сторінках, тому зовнішній вигляд елемента достатньо описати один раз.

Об'єкти на сторінці розміщуються за допомогою HTML. А ось CSS відповідає за те, як ці об'єкти виглядають. Їх розмір, колір, фонове зображення, рівень прозорості, розташування щодо інших елементів, поведінка при наведенні курсору, візуальна зміна кнопок при натисканні тощо.

Щоб реалізувати більш складні процеси потрібно використовувати мову JavaScript, яка є однією з найпоширеніших мов у веб-розробці. JavaScript – це універсальна об’єктно-орієнтована мова програмування. Ця мова найчастіше використовується в розробці програм для браузерів, щоб надати їм інтерактивність і динамічності.

Сучасний JavaScript є «безпечною» мовою програмування. Він не надає низькорівневого доступу до пам’яті або ЦП, оскільки спочатку був розроблений для браузерів, яким він не був потрібен. Функціональність JavaScript залежить від середовища, в якому він працює. Використання JavaScript обумовлене тим, що для роботи з цією мовою не потрібно щось завантажувати, усе потрібне знаходиться у будь-якому браузері.

Після ознайомлення з технологіями та середою розробки для створення веб-додатку і налаштування мережі компанії в емуляторі Cisco Packet Tracer, можна приступити до розробки веб-додатку.

3.2 Функціональний огляд розробленого веб-додатку

Розроблений веб-додаток являє собою два блоки. Перший блок складається з мережі яка була створена в емуляторі Cisco Packet Tracer (рис. 3.1). Зображення мережі використовується для полегшення налаштування розширених списків контролю доступу, коли користувач може бачити усю мережу, він краще розуміє які списки йому потрібно налаштувати.

Другий блок знаходиться з правої сторони від схеми, в ньому знаходяться елементи які потрібно заповнити, щоб отримати бажані налаштування списків контролю доступу.

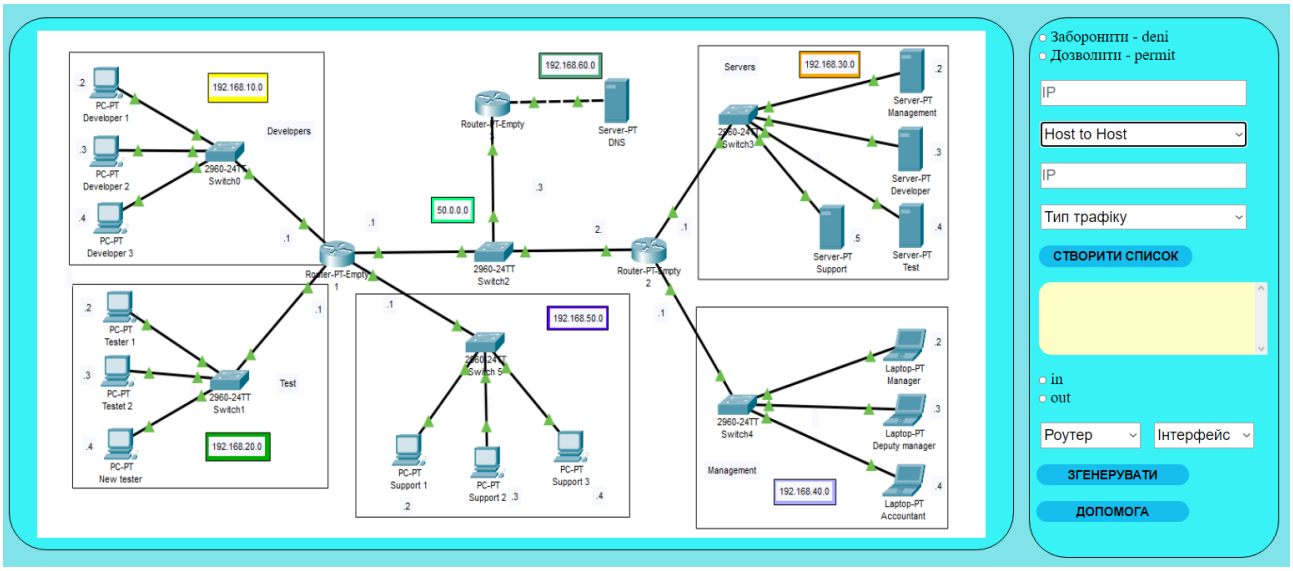

Рисунок 3.1 – Інтерфейс веб-додатку

В блоці два знаходиться весь функціонал додатку (рис 3.2). Функціонал складається з пустих полів, кнопок, текстового поля і перемикачів. Завдяки перемикачам «Заборонити» та «Дозволити» можна обрати, що користувач хоче робити з конкретними типами трафіку.

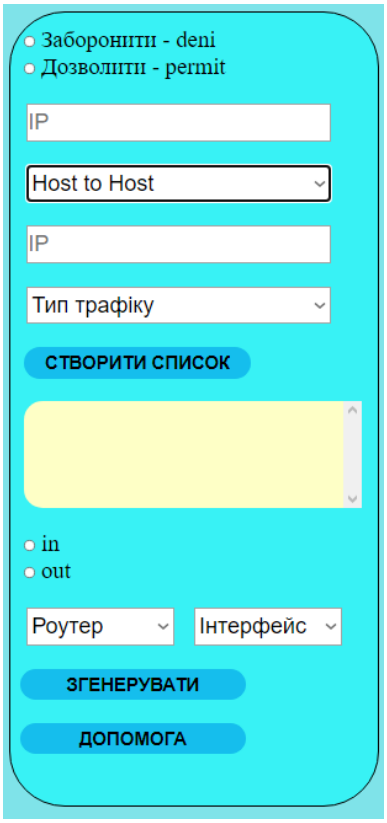

Рисунок 3.2 – Блок налаштування ACL

Натиснувши на випадаючий список під назвою “Оберіть тип ALC” з’явиться чотири варіанта на вибір «Хост до хосту», «Хост до мережі», «Мережа до хосту» і «Мережа до мережі». У випадку якщо користувач обере тип який містить в своїй назві “Network” то йому буде необхідно ввести ще й обернену маску для адреси джерела або призначення (рис 3.3).

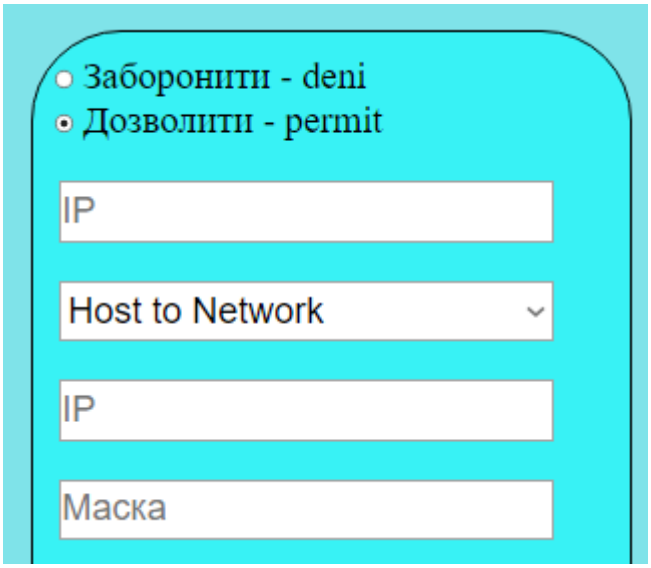

Рисунок 3.3 – Залежність відображення полів і від типу ACL

Якщо натиснути на випадаючий, що має назву “Тип трафіку” то користувач зможе обрати тип трафіку який він хоче налаштувати в своєму ACL, на вибір три найпопулярніших види трафіку “ICMP”, “TCP” і “IP”. Якщо вибрати тип трафіку “TCP” то користувачу потрібно буде вибрати один з двох портів (рис. 3.4).

Рисунок 3.4 – Вибір порта після вибору TCP-трафіка

Після того, як користувач ввів усі необхідні дані та обрав усі потрібні налаштування, він повинен натиснути кнопку «Додати правило». У вікні нижче з’явиться сформоване правило (рис 3.5).

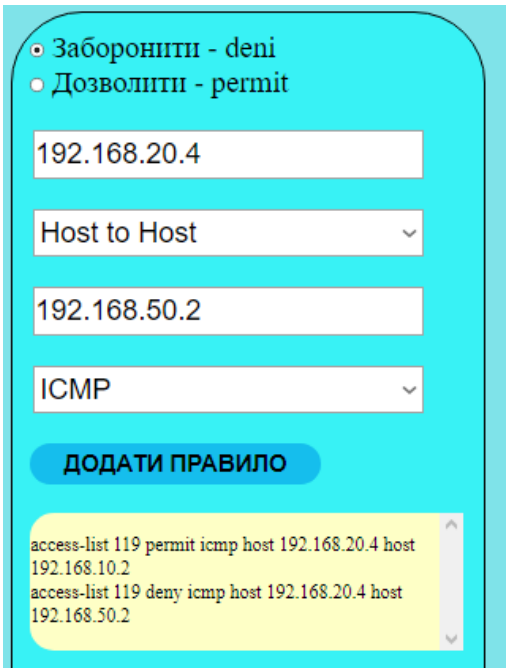

Рисунок 3.5 – Вигляд згенерованого списку правил

Після того, як користувач додав усі необхідні правила, йому потрібно вибрати маршрутизатор, до якого він хоче застосувати список доступу, вибрати інтерфейс цього маршрутизатора та вхідний або вихідний трафік (рис 3.6).

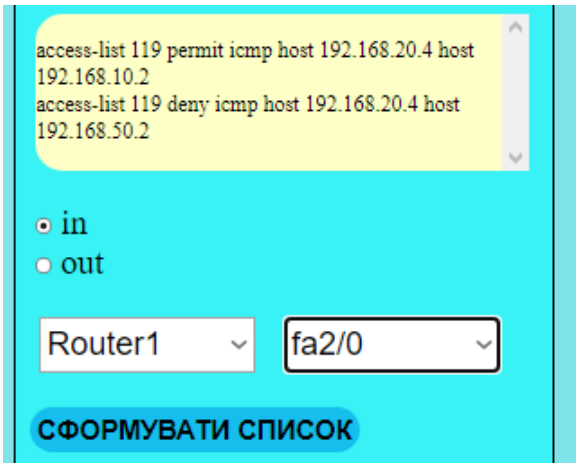

Рисунок 3.6 – Налаштування параметрів напрямку, маршрутизатора та інтерфейсу

Остання дія, яку треба виконати користувачу, це натиснути на кнопку “Сформувати список”, як тільки він натисне на кнопку з’явиться вікно (рис. 3.7). Щоб скопіювати код потрібно натиснути кнопку “Скопіювати”(рис. 3.8).

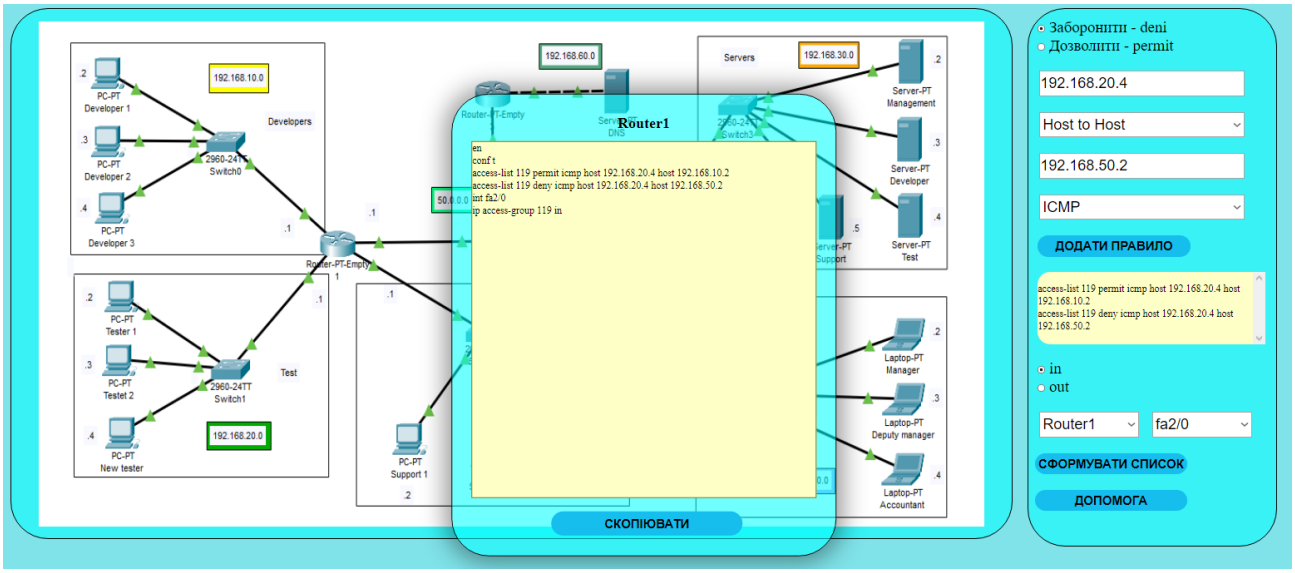

Рисунок 3.7 – Зовнішні вигляд інтерфейсу після натискання кнопки

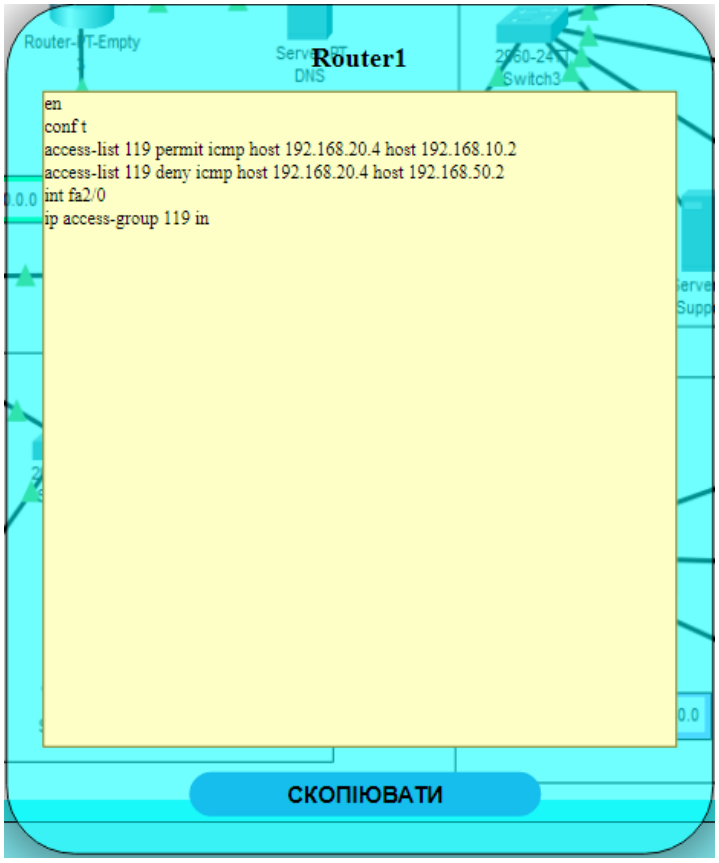

Рисунок 3.8 – Вигляд коду готового до копіювання

Для користувачів які не вміють налаштовувати ACL є кнопка “Допомога”, після натискання кнопки з’явиться окреме вікно з інструкцією як налаштувати ACL за допомогою веб-додатка (рис 3.9).

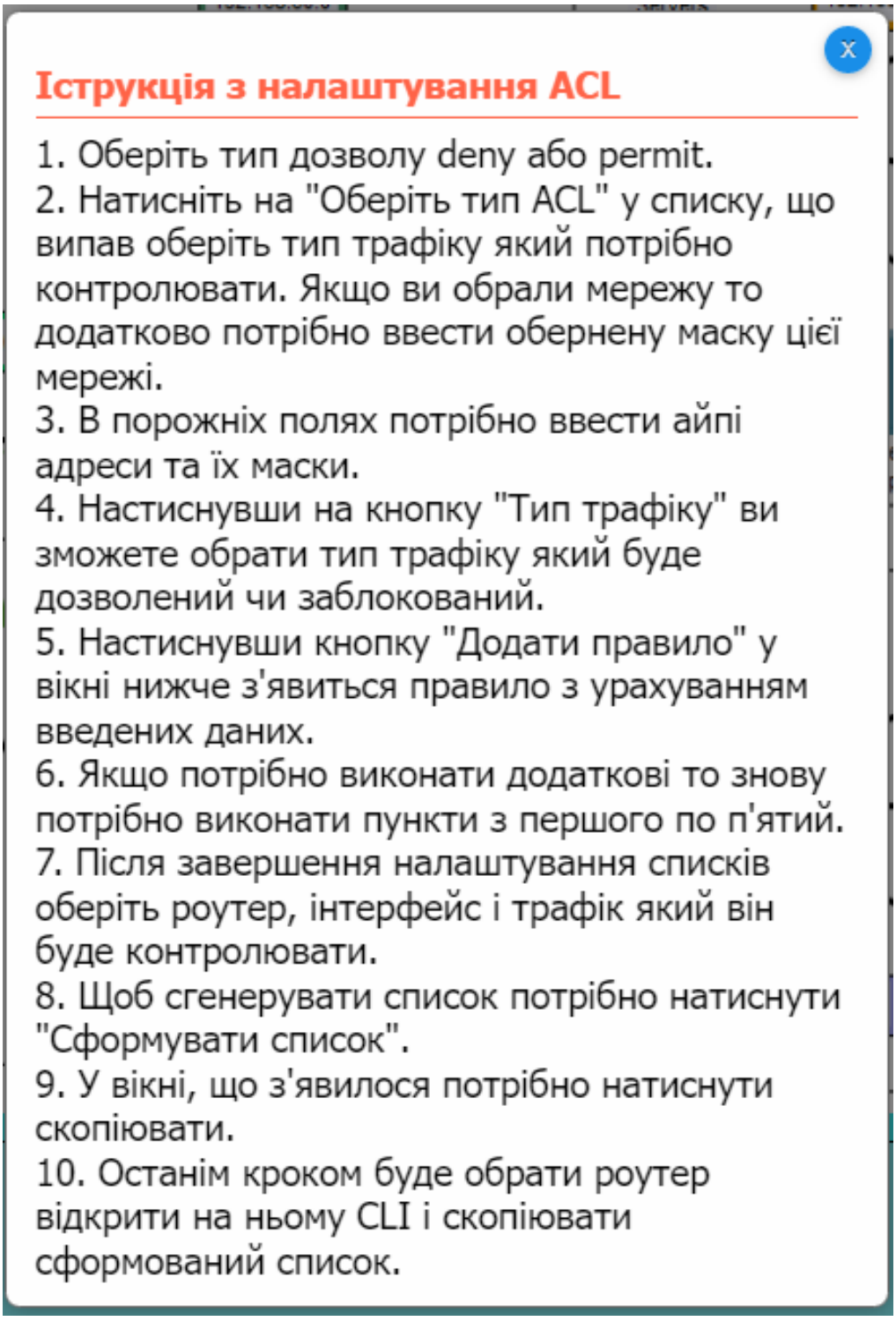

Рисунок 3.9 – Інструкція за налаштування ACL за допомогою веб-додатку

3.3 Тестування веб-додатку автоматичної генерації списків в Cisco Packet Tracer

Щоб переконатися, що веб-додаток генерує правильні списки, потрібно провести випробування в новій мережі створеної в емуляторі Cisco.Нова мережа буде схожа на першу, але більш спрощена (рис. 3.10). В спрощеній мережі немає DNS сервера і відділу підтримки.

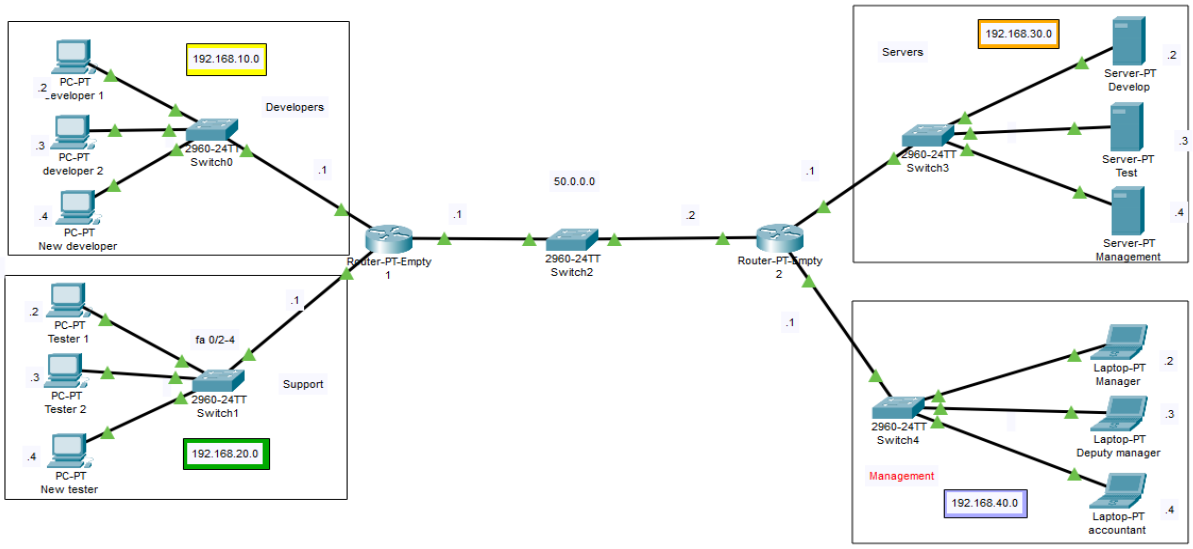

Рисунок 3.10 – Вигляд нової мережі

По-перше потрібно налаштувати на маршрутизаторі 1 обмеження для робітника New tester, щоб він не міг пінгувати нікого у мережі розбронників, окрім робітника developer 1.

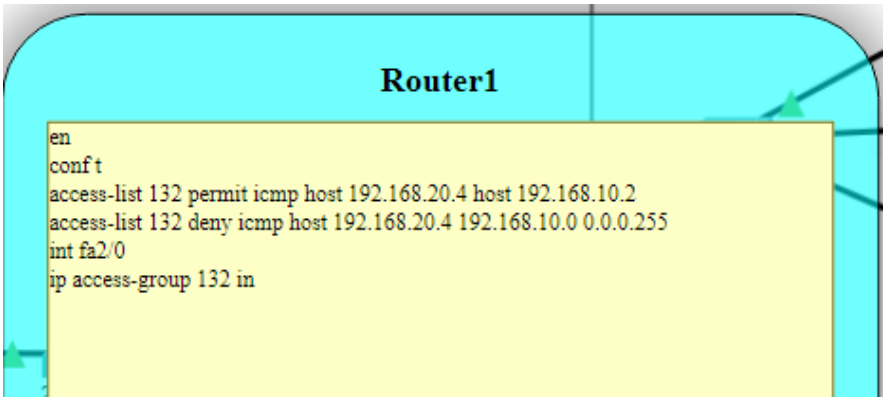

Рисунок 3.11 – Сформований список для тестування трафіку Host to Host та Host to Network

Як видно на рисунку 3.11 спочатку був налаштований дозвіл на отримання трафіку від робітника New tester до робітника developer 1, а другим правилом заборонили мати трафік з іншими користувачами мережі розброників.

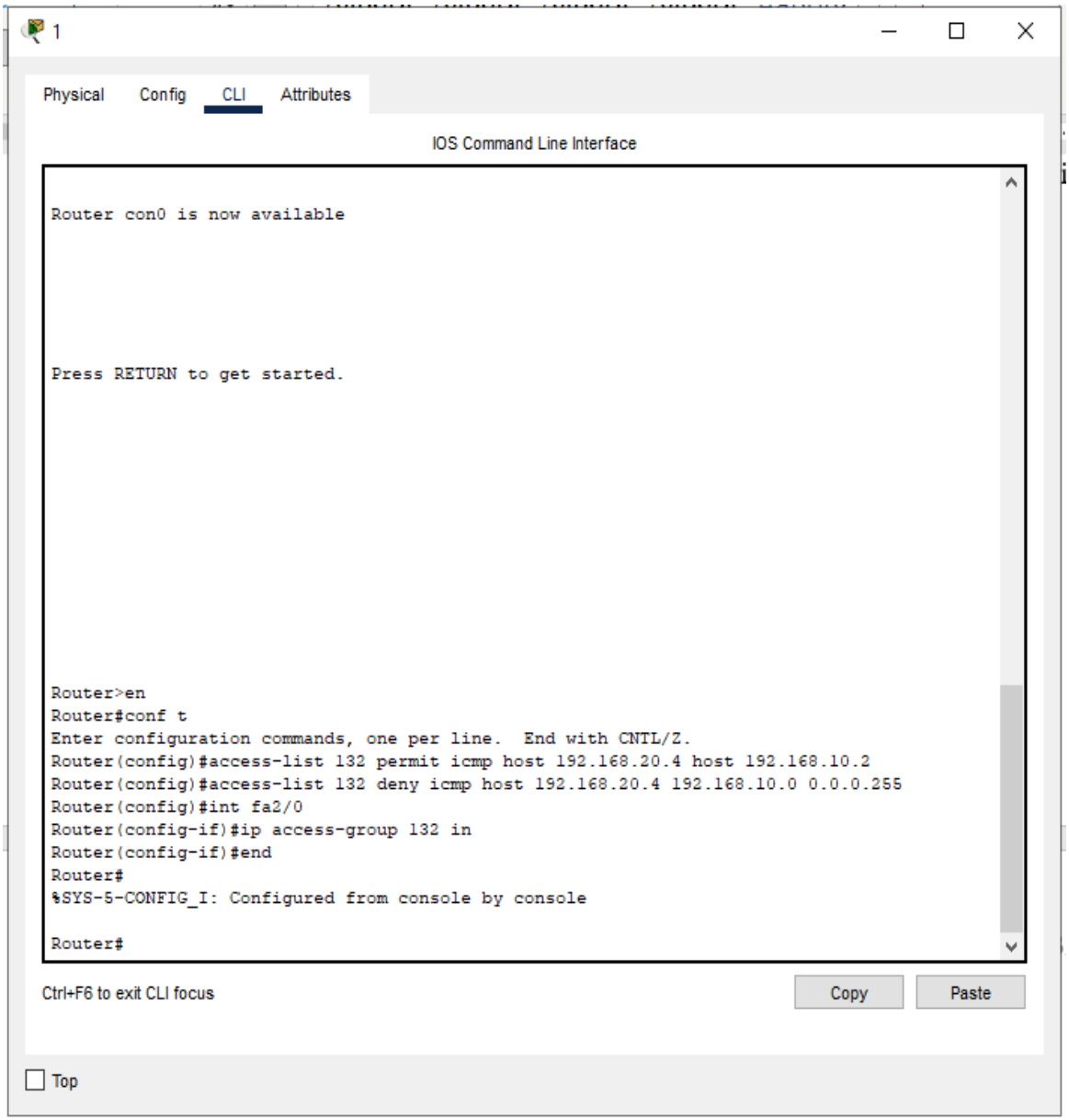

Рисунок 3.12 – Налаштування маршрутизатор 2 використовуючи створений ACL

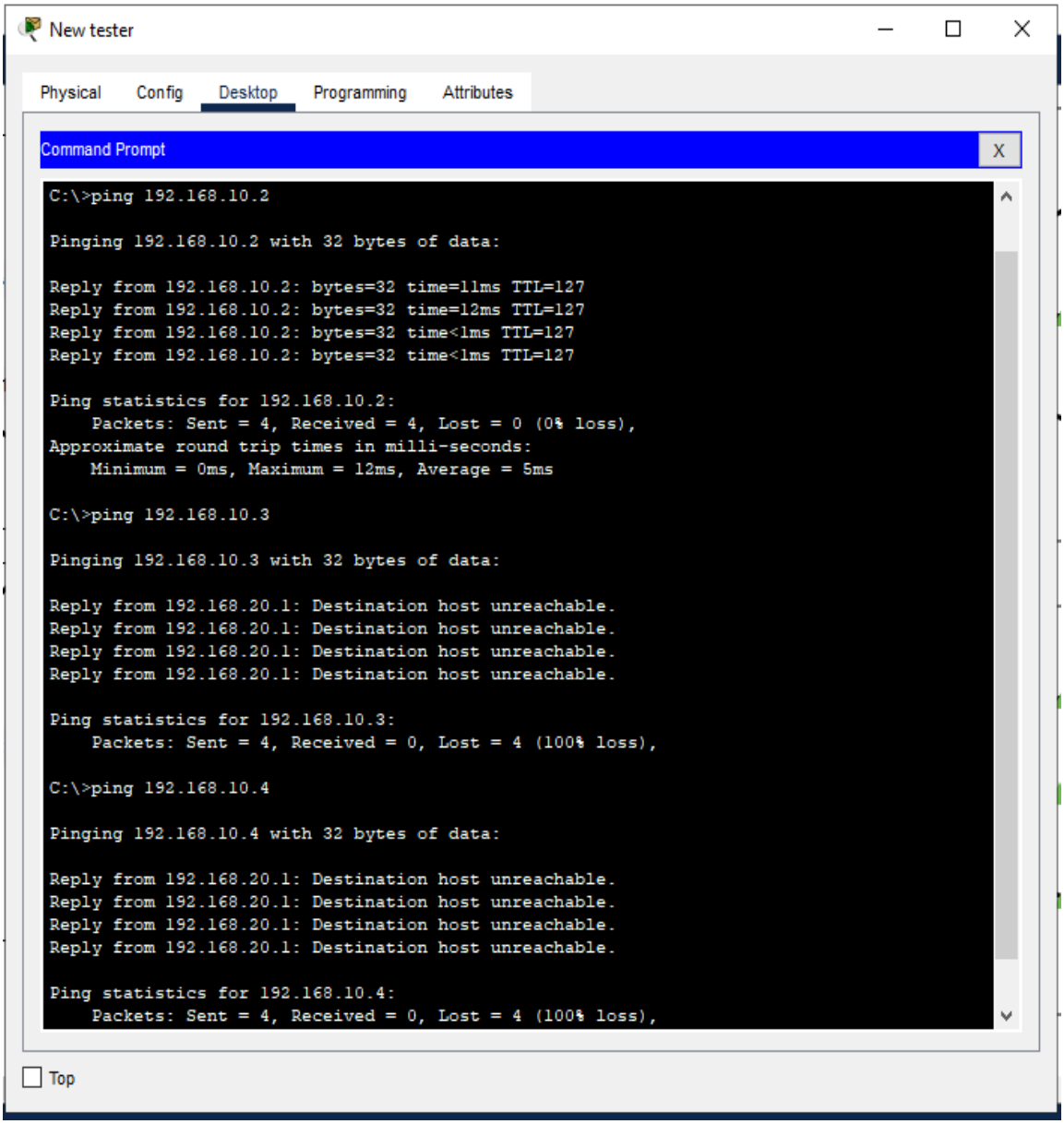

Рисунок 3.13 – Перевірка правильності роботи ACL

На рисунку 3.13 можна побачити, що дозволений трафік йде туди, куди йому потрібно, а заборонений трафік був знищений, це означає, що перевірка працездатності ACL створених веб-додатком пройшла успішно.

Тепер заблокуємо доступ з мережі тестувальників та розробників до мережі менджерів (рис 3.14), це потрібно для перевірки коректності роботи з масками під час генерації правил, які направлені на роботу з мережами.

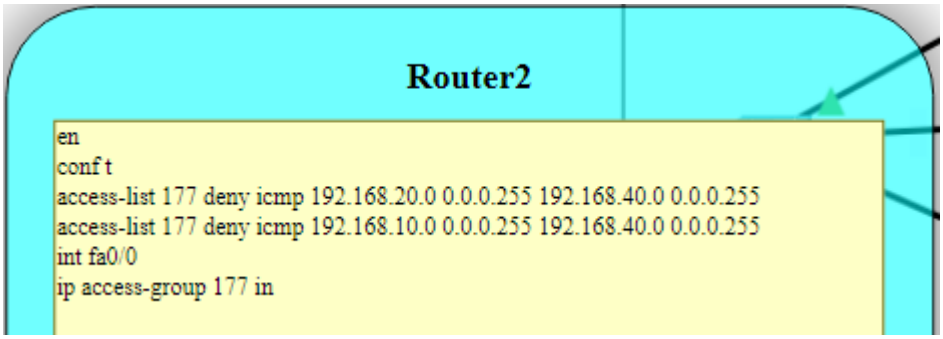

Рисунок 3.14 – Сформований список для тестування Network to Network

Копіюємо отриманий ACL і вставляємо його в CLI на маршрутизаторі 2 (рис 3.15).

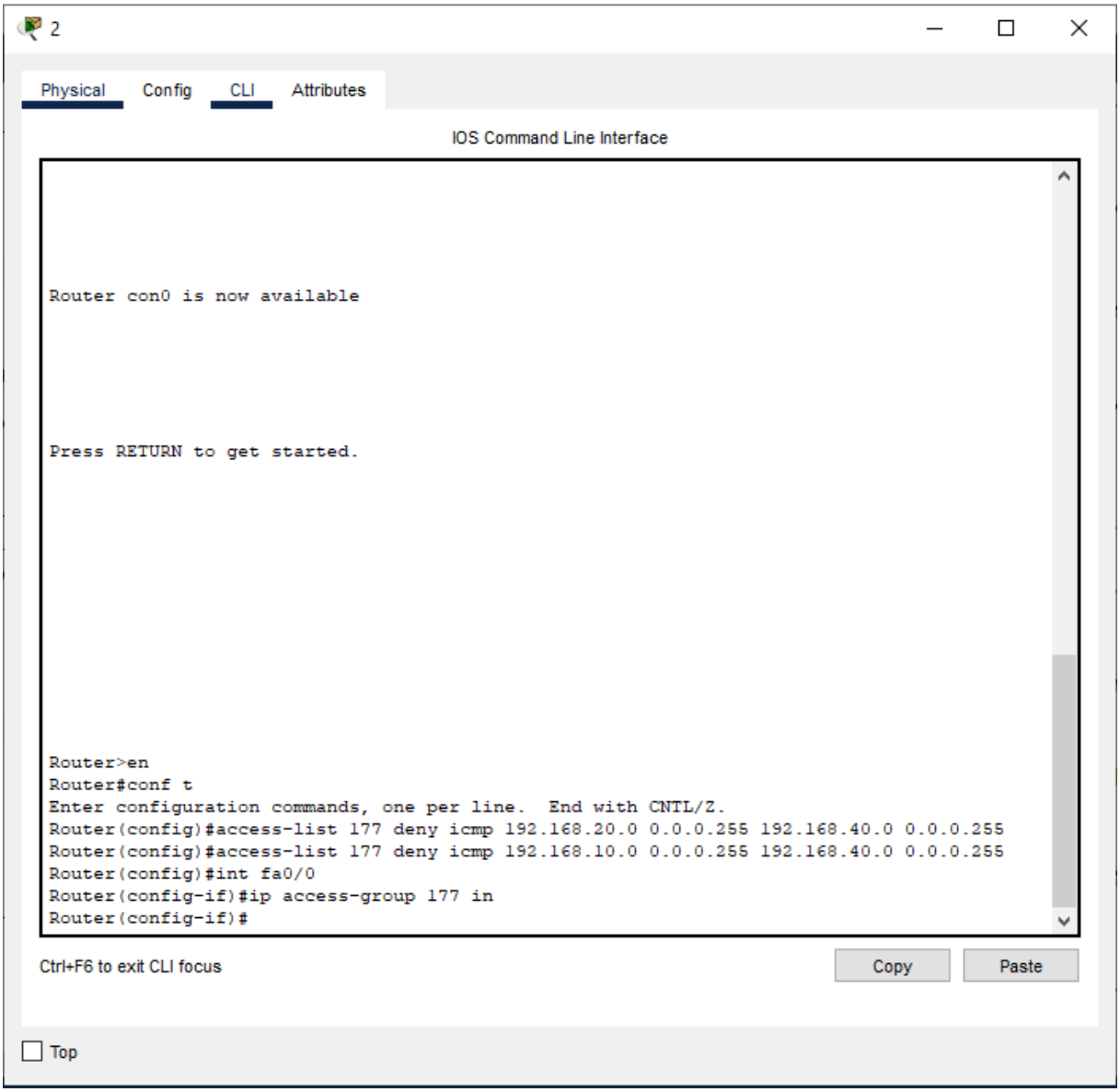

Рисунок 3.15 – Налаштування маршрутизатор 2 використовуючи створений ACL

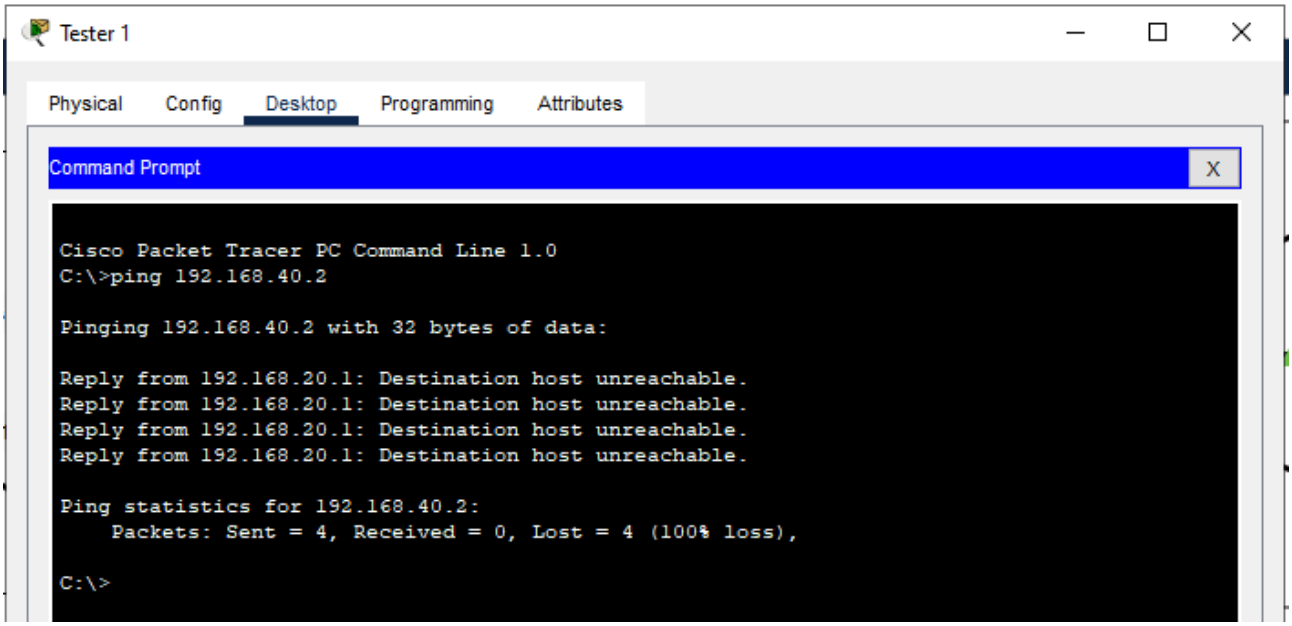

Рисунок 3.16 – Знищення пакетів з мережі тестувальників

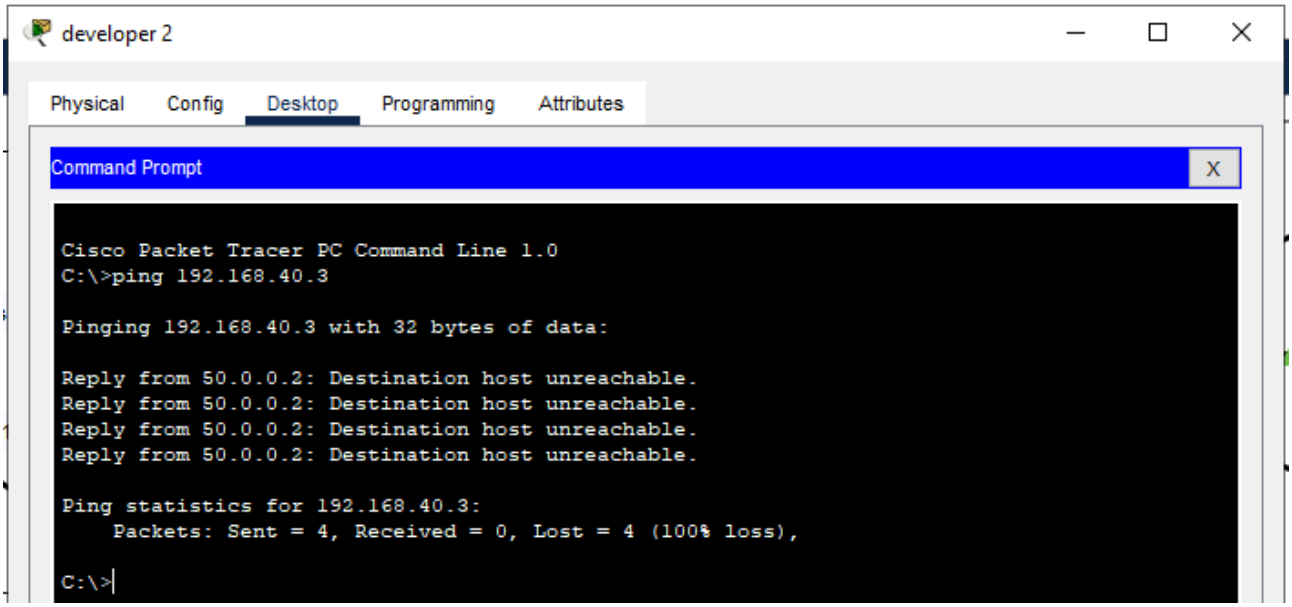

Рисунок 3.17 – Знищення пакеті з мережі розробників

Останнім тестом будет перевірка, як працює генерація коду розширеного списку доступу, якщо в ньому є правило, що забороняє tcp-трафік. Оскільки в мережі немає DNS, доступ до серверів тепер не захищений, тому потрібно налаштувати заборону відправки ftp-пакетів на сервер, але потрібно залишити доступ для http-трафіку. Для прикладу буде заборонено ftp-трафік на сервер менджменту і дозвіл на http-трафік (рис 3.18).

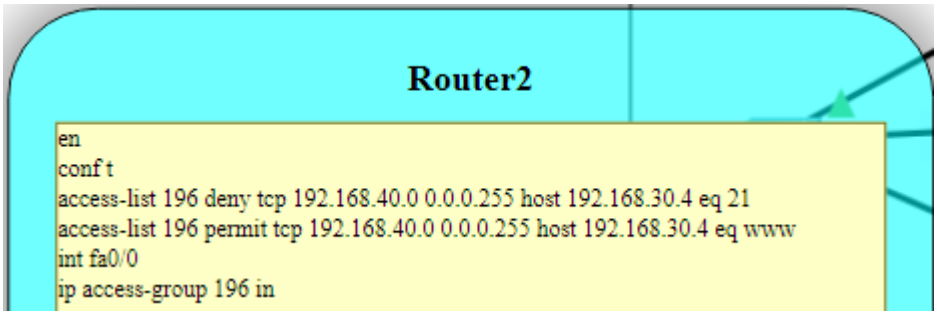

Рисунок 3.18 – Сформований список для тестування налаштувань TCP

Копіюємо отриманий ACL і вставляємо його в CLI на маршрутизаторі 2 (рис 3.19).

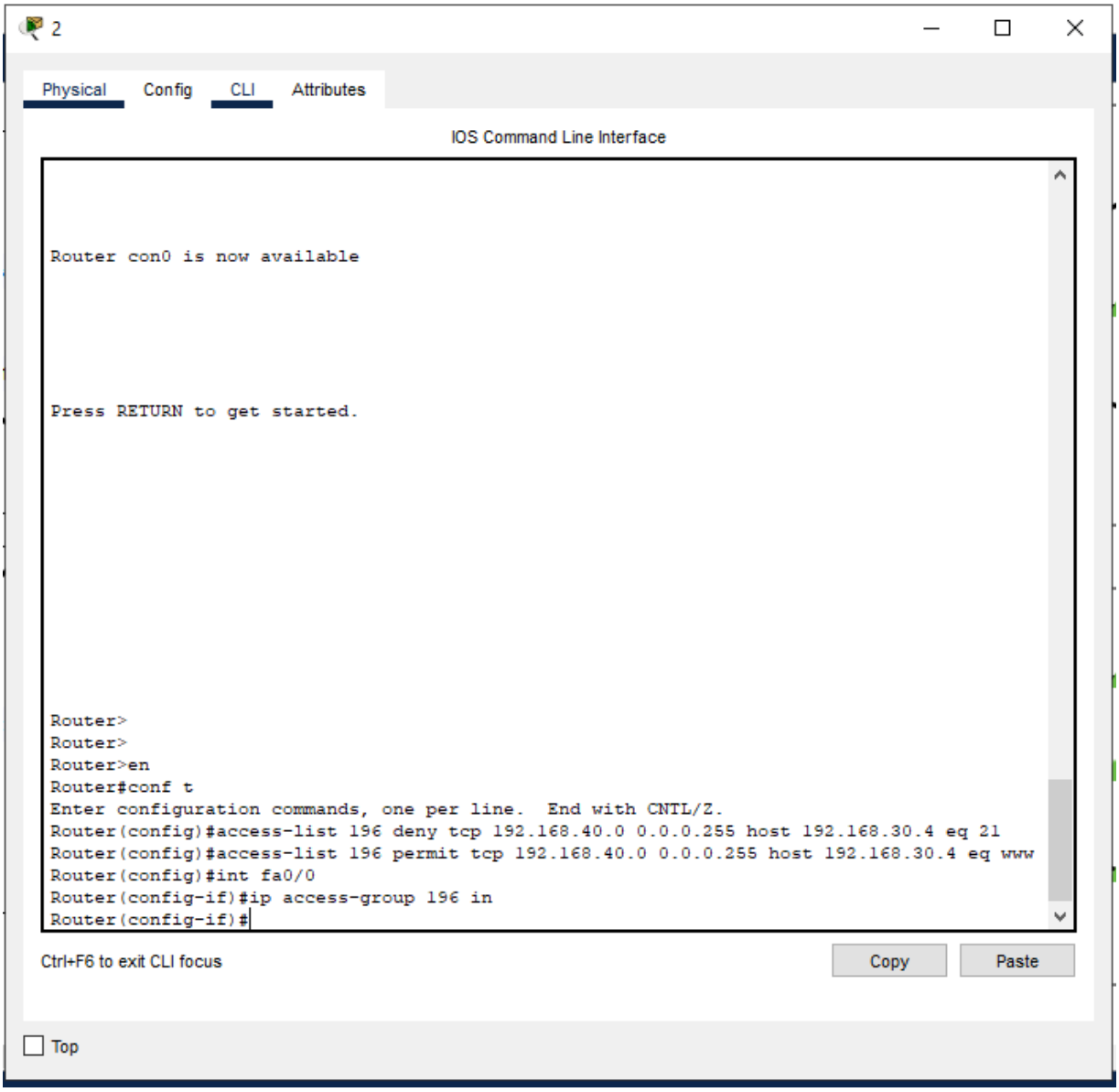

Рисунок 3.19 – Налаштування маршрутизатора 2 використовуючи створений ACL

Тепер потрібно створити ftp-пакет та http-пакет(рис3.20,3.21) і спробувати їх відправити на сервер менеджерів.

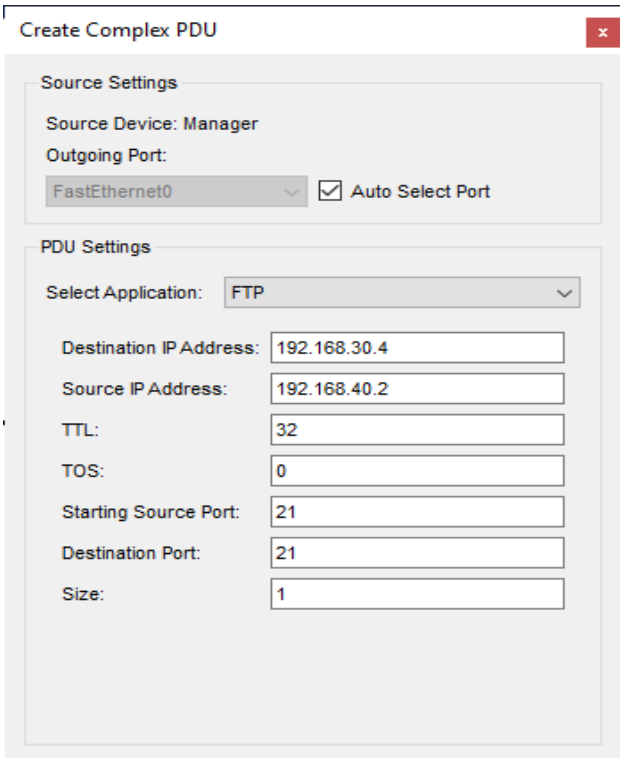

Рисунок 3.20 – Створення ftp -пакету

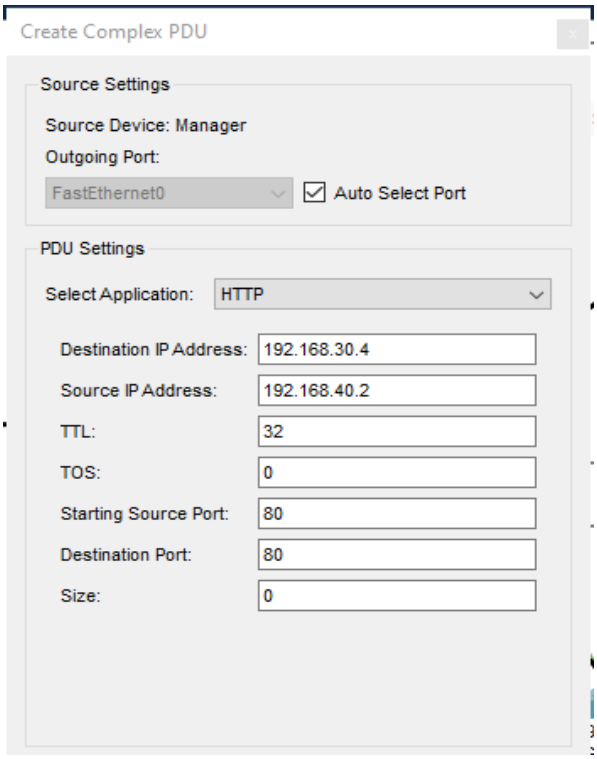

Рисунок 3.21 – Створення http -пакету

В результаті отримуємо , що ftp-пакет заблокований, а http-пакет ні. Для більшої впевненості був ще один тест з іншого комп’ютера результати на рисунку 3.22.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num
	Failed	Manager	192.168.30.4	TCP		1.000	N	0
	Successful	Manager	192.168.30.4	TCP		1.000	N	1
	Failed	Deputy manager	192.168.30.4	TCP		1.000	N	2
	Successful	Deputy manager	192.168.30.4	TCP		1.000	N	3

Рисунок 3.22 – Результат заборони ftp-трафіку та дозволу http-трафіку

У результаті тестування веб-додатку автоматичної генерації списків доступу було виявлено, що усі налаштування працюють як треба. Отже це значить, що навіть не обізнаний користувач за допомогою інструкції швидко і якісно зможе налаштувати ACL списки.

ВИСНОВКИ

Виходячи з цілей, сформованих на початку роботи, можна зробити наступні висновки:

В роботі були проаналізовані усі функції та особливості ACL списків. Завдяки використанню розширених списків контролю доступу вдалося створити надійну, оптимізовану, а головне захищену мережу .

За допомогою емулятора Cisco Packet Tracer була розроблена мережа в якій дуже добре продемонстровано, як розширені ACL впливають на трафік в середині мережі. В результаті роботи з мережею були сформовані базові правила роботи з ACL, які у майбутньому були використані в розробці системи автоматичної генерації списків.

В рамках роботи був розроблений веб-додаток, який в значній мірі спрощує налаштування розширених ACL. Результатом роботи з веб-додатком є швидке і якісне налаштування розширених списків контролю доступу. Додаток використовує найпоширеніші види налаштувань ACL.

Даним веб-додатком може користуватися хто завгодно в незалежності від рівня його обізнаності в налаштуванні ACL. Для початківців додаток містить інструкцію в якій описано як налаштувати розширені ACL, користуючись ним.

СПИСОК ЛІТЕРАТУРИ

1. Організація комп’ютерних мереж: підручник: для студ. спеціальності 122 «Комп’ютерні науки» / КПІ ім. Ігоря Сікорського ; Ю. А. Тарнавський, І. М. Кузьменко. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.

2. ACL: списки контроля доступа в Cisco IOS [Електронний ресурс] – https://habr.com/ru/post/121806/

3. Списки контроля доступа [Електронний ресурс] – https://intuit.ru/studies/courses/3646/888/lecture/31159?page=5

4. Сетевые технологи ACL и NAT [Електронний ресурс] – https://ppt-online.org/1150817

5. Access Control List Explained with Examples [Електронний ресурс] – https://www.computernetworkingnotes.com/ccna-study-guide/access-control-list- explained-with-examples.html

6. Configure Commonly Used IP ACLs [Електронний ресурс] – https://www.cisco.com/c/en/us/support/docs/ip/access-lists/26448-ACLsamples.html

7. Что такое ACL и как его настраивать [Електронний ресурс] – http://ciscotips.ru/acl

8. Configuring extended ACLs [Електронний ресурс] – https://study-ccna.com/configuring-extended-acls/

9. Списки доступа ACL. Настройка статического и динамического NAT [Електронний ресурс] – https://intuit.ru/studies/courses/3549/791/lecture/29226?page=2

10. Access Control List Explained with Examples [Електронний ресурс] – https://www.computernetworkingnotes.com/ccna-study-guide/access-control-list-explained-with-examples.html

11. Cisco Packet Tracer: что нам стоит сеть построить? [Електронний ресурс] – https://www.cisco.com/c/dam/global/ru_ua/assets/pdf/cisco_packet_tracer.pdf

12.Type of Access Control Lists [Електронний ресурс] – https://www.learncisco.net/courses/icnd-1/acls-and-nat/type-of-acls.html

ДОДАТОК А

Client.js

```
const create_button = document.querySelector("#create_button");
const unic_button = document.querySelector("#unic_button");
const box = document.querySelector(".box");
const toggle = () => {
    if (box.style.display != "flex") {
        box.style.display = "flex";
    } else {
        box.style.display = "none";
    }
    console.log(box.style.display)
};

const unic_acl_type = document.querySelector("#aclType");
const radio_buttons = document.getElementsByName('per_or_deny');
const traffic_t = document.querySelector("#traffic_unic");
const unic_port_type = document.querySelector('#unic_port');
const new_button = document.querySelector("#add_rule");
const rules_list = document.querySelector('.create_rule');
const ip_source = document.querySelector("#unic_ip");
const firstIPMaskField = document.querySelector("#mask1");
const secondIPMask = document.querySelector("#mask2");
const dest_ip = document.querySelector("#unic2_ip");
let source_ip, ip_dest, acl_type, step, traffic_type, port_type;
let unic_num = Math.round(Math.random() * (199 - 100) + 100);
let new_rule = 'access-list ' + unic_num + ' ';
const new_rules = [];

const create_acl_list = () => {
    let content, input_data, i;
    content = document.querySelector(".tool_ip");
    input_data = content.getElementsByTagName('input');
    for (i = 0; i < input_data.length; ++i) {
        if (input_data[i].type == "text")
            input_data[i].value = '';
    }
}
traffic_t.addEventListener("change", () => {
    unic_port_type.style.display = traffic_t.value == "tcp" ? 'block' : 'none'
})
```

```
unic_acl_type.addEventListener("change", () => {
    let status = unic_acl_type.value;
    if (status == 0) {
        firstIPMaskField.style.display = "none";
        secondIPMask.style.display = "none";
    }
    if (status == 1) {
        firstIPMaskField.style.display = "none";
        secondIPMask.style.display = "block";
    }
    if (status == 2) {
        firstIPMaskField.style.display = "block";
        secondIPMask.style.display = "none";
    }

    if (status == 3) {
        firstIPMaskField.style.display = "block";
        secondIPMask.style.display = "block";
    }
});
new_button.addEventListener("click", () => {
    if (ip_source.checkValidity() && dest_ip.checkValidity()) {
        source_ip = ip_source.value;
        ip_dest = dest_ip.value;
        acl_type = unic_acl_type.value;
        port_type = unic_port_type.value;
        for (let i = 0, length = radio_buttons.length; i < length; i++)
            if (radio_buttons[i].checked) {
                step = radio_buttons[i].value;
                break;
            }
        traffic_type = traffic_t.value;
        let source_type, dest_type;
        switch (acl_type) {
            case "0":
                source_type = "host " + source_ip;
                dest_type = "host " + ip_dest;
                break;
            case "1":
                source_type = "host " + source_ip;
                dest_type = ip_dest + " " + secondIPMask.value;
```

```
                break;
            case "2":
                source_type = source_ip + " " + firstIPMaskField.value;
                dest_type = "host " + ip_dest;
                break;
            case "3":
                source_type = source_ip + " " + firstIPMaskField.value;
                dest_type = ip_dest + " " + secondIPMask.value;
                break;
        }
        new_rule += step + " " + traffic_type + " " + source_type + " " + dest_type;
        if (traffic_t.value == "tcp") new_rule += " " + port_type;
        new_rules.push(new_rule);
        ruleView = document.createElement("span");
        ruleView.textContent = new_rule;
        rules_list.appendChild(ruleView);
        console.log(ruleView);
        console.log(new_rules);
        new_rule = "access-list " + unic_num + " ";
        create_acl_list();
    } else {
        errorMessage = "Невірна IP-адреса";
        ip_source.setCustomValidity(errorMessage);
        dest_ip.setCustomValidity(errorMessage);
        ip_source.reportValidity()
        dest_ip.reportValidity()
    }
});
const box_text = document.querySelector('.box_text')
create_button.addEventListener("click", () => {
    toggle();
    document.querySelector('.box_header').innerHTML = document.getElementById('select_router').value
    new_rules.forEach(new_rule => {
        box_text.innerHTML += new_rule + '<br>'
    });
    box_text.innerHTML += 'select_interface ' + document.getElementById('select_interface').value + '<br>'
    let directionsRadio = document.getElementsByName('direction');
    let direction = "";
    for (let i = 0, length = directionsRadio.length; i < length; i++)
        if (directionsRadio[i].checked) {
            direction = directionsRadio[i].value;
```

```
            break;
        }
    box_text.innerHTML += "ip access-group " + unic_num + " " + direction
});
const copy_acl = str => {
    const copy_doc = document.createElement('textarea');
    copy_doc.value = str;
    document.body.appendChild(el);
    copy_doc.select();
    document.execCommand('copy');

};
unic_button.addEventListener("click", () => {
    copy_acl(box_text.innerText)
    box_text.innerHTML = "enable <br> configure terminal <br>"
    toggle();
});
```