МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра комп'ютерної інженерії

Пліш Вадим Володимирович

Програмний модуль аналізу конфігурації маршрутизатора / Router configuration analysis software module

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна

Виконав: студент групи КІз-41
Пліш Вадим Володимирович

Науковий Керівник
к.т.н. Мельник Г.М.

ТЕРНОПІЛЬ-2023

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний модуль аналізу конфігурації маршрутизатора» зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «бакалавр» містить 67 сторінок пояснюючої записки, 19 рисунків, 2 таблиці, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розроблення програмного модуля аналізу конфігурації маршрутизатора. Конфігурація маршрутизатора це інформація про налаштування мережевих інтерфейсів (включаючи IP-адреси, підмережі, маски підмереж, VLAN), маршрутних таблиць, правила мережевого екрану (правила фільтрації, перенаправлення портів, NAT).

Методи дослідження включають методи фізичної і логічної структуризації комп’ютерних мереж, методи об’єктного програмування.

Розроблено об'єктну модель для представлення правил фільтрації і специфічні класи для різних пристроїв. Кожен клас має свої атрибути і методи, що відповідають програмній моделі відповідного пристрою. Схема правил маршрутизаторів варіюється залежно від класу маршрутизатора, але має деякі спільні компоненти. Основними компонентами структури правила для фільтрації трафіку є умова, дія, пріоритет, напрямок, джерело та призначення.

Розроблено алгоритми аналізу конфігурації на основі команд ОС RouterOS для налаштування інтерфейсів, маршрутизації, файрволу, безпеки розроблено функції. Для дистанційного керування пристроєм і витягнення конфігурації використано захищений протокол SSH. Програмно реалізовано розроблені алгоритми на мові Python.

Ключові слова: маршрутизатор, правило фільтрації, SSH.

RESUME

Qualification thesis “Router configuration analysis software module” in the specialty 123 "Computer Engineering" of bachelor education degree contains 67 pages of explanatory notes, 19 figures, 2 tables, 3 appendixes. The volume of graphic material is 2 sheets of A3 format.

The aim of the qualification work is to develop a software module for analyzing router configuration. Router configuration includes information about the settings of network interfaces (including IP addresses, subnets, subnet masks), routing tables, and network firewall rules (filtering rules, port forwarding, NAT).

The research methods include methods of physical and logical structuring of computer networks and object-oriented programming methods.

An object model has been developed to represent filtering rules, and specific classes have been created for different devices. Each class has its own attributes and methods that correspond to the software model of the respective device. The schema of router rules varies depending on the router class but has some common components. The main components of the traffic filtering rule structure are condition, action, priority, direction, source, and destination.

Algorithms for analyzing configuration based on RouterOS commands have been developed to configure interfaces, routing, firewall, and security functions. A secure SSH protocol has been used for remote device management and configuration retrieval. The developed algorithms have been implemented in Python programming language.

Please note that translations can sometimes be subjective, and it's recommended to review and adapt them based on the specific context.

Keywords: ROUTER, FILTERING RULE, SSH.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ME	–	Міжмережевий екран
VLAN	–	Virtual Local Area Network
VPN	–	Virtual Private Network
SNMP	–	Simple network management protocol
MIB	–	Management Information Base

ВСТУП

Проблема інформаційної безпеки в корпоративних мережах зв'язку сьогодні дуже гостро стоїть перед компаніями будь-якого рівня. Втрата критично важливої корпоративної інформації, ріст обсягів паразитного трафіка, вимагання, шантаж і замовлені атаки на інформаційні ресурси підприємства стали частим явищем. Важливість інтернету для будь-якого сучасного підприємства - зрозуміла й незаперечна. Інтернет служить для комунікації з партнерами й замовниками (у тому числі через корпоративний веб-сайт), забезпечення дистанційного доступу до корпоративних ресурсів і гнучкого розширення робочого простору, пошуку корисної інформації й здійснення електронних комерційних транзакцій. Тим часом інтернет має рядом властивостей, які утрудняють забезпечення інформаційної безпеки:

- інтернет - це публічна відкрита мережа з нецентралізованими топологією й маршрутизацією;
- шкідлива активність може виникнути в одній частині інтернету й потім швидко поширитися по всій Всесвітній мережі;
- в інтернеті контролюється головним чином трафік, що входить, але не вихідний;
- у всесвітній мережі практично відсутня ідентифікація користувачів;
- юрисдикція країни, у якій відбувся злочин, найчастіше не поширюється на кіберзлочинця.

Маршрутизатори та правила фільтрації трафіку є важливими складовими мережевої безпеки. Розробка програмних засобів, що дозволяють аналізувати конфігурацію маршрутизатора та правила фільтрації трафіку, допомагає виявляти потенційні проблеми безпеки, такі як вразливості, помилки конфігурації або некоректні правила фільтрації.

Аналіз конфігурації маршрутизатора дозволяє зрозуміти, як він налаштований і які маршрути та політики розподілення трафіку

використовуються. Це може допомогти виявити недоліки, зайві або некоректні налаштування, які можуть призводити до незадовільної продуктивності мережі або перевантаження.

Аналіз правил фільтрації трафіку дозволяє виявити проблеми зі згортанням мережі, некоректною обробкою пакетів або забороненими або небажаними правилами фільтрації. Це допомагає виявити потенційні проблеми, такі як маршрутизаційні петлі, блокування певного виду трафіку або недосяжність певних мереж [1-3].

Отже актуальним є розроблення програмного засобу для візуалізації налаштувань і правил фільтрації на третьому мережевому рівні моделі OSI.

Об’єкт дослідження – міжмережеві екрани. Предмет дослідження – правила фільтрації трафіку. Метою кваліфікаційної роботи є розроблення програмного модуля аналізу конфігурації і правил фільтрації трафіку маршрутизатора. Для досягнення мети такі завдання:

– проаналізувати методи логічної структуризації мереж;

– проаналізувати функції протоколів маршрутизації та мережевих фільтрів;

– розробити алгоритми аналізу налаштувань мережевих інтерфейсів;

– розробити алгоритми аналізу маршрутних таблиць;

– програмно реалізувати розроблені алгоритми з метою отримання правил фільтрації трафіку;

– здійснити тестування програмного засобу.

За результатами роботи опубліковано тези доповіді на VII науково-практичній конференції «Інтелектуальні комп’ютерні системи та мережі» [1]. Копії публікації наведено у додатку В.

В третьому розділі розроблено UML діаграми і програмно реалізовано програмного забезпечення.

1 ФУНКЦІЇ І ПРИЗНАЧЕННЯ МАРШРУТИЗАТОРІВ

1.1 Види маршрутизаторів

Маршрутизатор є мережевим пристроєм, який приймає пакети даних з однієї мережі та вирішує, як переслати їх до іншої мережі. Він працює на рівні мережевого протоколу (рівень 3 OSI-моделі) та використовує інформацію з мережевих таблиць для визначення оптимального шляху для пересилання пакетів.

Міжмережевий екран (МЕ) - це пристрій або програмне забезпечення, яке використовується для фільтрації трафіку в мережі. Його основна мета полягає у захисті мережі від несанкціонованого доступу та зловмисних атак. МЕ може аналізувати заголовки пакетів, контролювати доступ до ресурсів та приймати рішення про пересилання або блокування пакетів на основі заданих правил.

Міжмережевий екран, є фільтром, який встановлюється між внутрішнім захищеним сегментом мережі та зовнішньою мережею, а також іншими сегментами Інтернету, і контролює всі потоки інформації всередині та навколо сегмента. Контроль трафіку здійснюється шляхом фільтрації, тобто вибіркового пропуску через екран, іноді з використанням спеціальних перетворень та створення повідомлень для відправника у разі відмови в доступі до його даних. Фільтрація базується на наборі заздалегідь завантажених у брандмауер умов, які відображають концепцію інформаційної безпеки підприємства. МЕ можуть бути реалізовані як апаратні або програмні комплекси, які встановлюються на комутаційні пристрої або сервери доступу вбудовані в операційну систему або працюють як програми, керовані цією системою [3-7].

МЕ зазвичай здійснюють такі функції:

– фізично відокремлюють робочі станції і сервери внутрішньої підмережі від зовнішніх каналів зв'язку;

– багатоетапно ідентифікують запити, що надходять до мережі;

– перевіряють повноваження і права доступу користувача до внутрішніх корпоративних ресурсів мережі;

– реєструють всі запити до компонентів внутрішньої підмережі ззовні;

– контролюють цілісність програмного забезпечення та даних;

– економлять адресний простір мережі (у внутрішній підмережі використовується локальна адресація);

– приховують IP-адреси корпоративних серверів для захисту від хакерів.

МЕ працюють на різних рівнях протоколів моделі OSI.

На мережевому рівні здійснюється фільтрація отриманих пакетів, що заснована на IP адресах (наприклад, не пропускати зовнішні пакети з Інтернету, спрямовані саме на ті сервери, доступ до котрих ззовні заборонений; не пропускати пакети з підробленими зворотніми адресами або IP адресами, доданми в «чорний список»). На четвертому транспортному рівні також можна фільтрувати за номерами портів TCP і прапорцями, що містяться в пакетах (запитів на встановлення нового з'єднання). На додатковому рівні може здійснюватись аналіз додаткових протоколів (HTTP чи SMTP) і контроль змісту потоків даних (заборона для внутрішніх абонентів отримувати будь-які типи файлів, наприклад, рекламної інформації чи виконуваних exe модулів).

Можна створювати в брандмауері експертну систему, яка, для аналізую трафіку, діагностує події, що можуть становити загрозу для безпеки внутрішньої мережі, і повідомляє про це адміністратора. Така експертна система також може автоматично посилювати умови фільтрування і т.д. у разі небезпеки (наприклад, спаму).

Апаратний МЕ діє як пристрій, що фізично підключається до мережі. Такий пристрій активно відстежує всі аспекти вхідного і вихідного обміну даними. МЕ перевіряє адреси джерела та призначення кожного оброблюваного повідомлення. Така дія допомагає забезпечити безпеку, що дозволяє ефективно запобігати небажаним проникненням у мережу або на вузол-комп'ютер. Зі свого боку, програмний брандмауер виконує аналогічні функції, але використовує встановлену на комп'ютері програму замість зовнішнього пристрою. Такий

підхід забезпечує аналогічний рівень безпеки та контролю над мережевими активностями.

Класифікація міжмережевих екранів заснована на відповідному рівні мережної моделі OSI [2, 8-15]. Розрізняють такі типи:

1. керовані комутатори (канальни рівень);

2. фільтри мережевого рівня (stateless filter) здійснюють аналіз IP-адреси відправника й приймача, протоколу та обидвох портів;

3. шлюзи мережевого рівня (circuit-level proxy), до таких шлюзів відносять ті що функціонують на умовному сеансовому рівні:

а. шлюзи трансляції адрес (NAT, PAT) або трансляції мережевих протоколів (транслюючі мости);

б. фільтри контролю стану каналу, або мережні фільтри з розширеними можливостями (stateful), які додатково аналізують і заголовки пакетів і фрагментовані пакети;

в. шлюзи мережевого рівня. Найбільш відомим і популярним шлюзом мережевого рівня є посередник SOCKS;

4. шлюзи прикладного рівня (application-level proxy) або проксі-сервери: на прозорі (transparent) і непрозорі (solid).

5. SPI міжмережевий екран (Stateful Packet Inspection) або пристрої з динамічною фільтрацією пакетів (Dynamic Packet Filtering), що є по суті шлюзами сеансового рівня з розширеними можливостями. Пристрої-інспектори стану оперують на сеансовому рівні OSI, але "розуміють" протоколи прикладного й мережного рівнів.

Існує також поняття "міжмережевий екран експертного рівня". Мережний екран даного типу базується на посередниках прикладного рівня або інспекторах стану, але обов'язково комплектується шлюзами сеансового рівня й мережними фільтрами, іноді розуміючи й мережний рівень. Найчастіше мають систему протоколювання подій і оповіщення адміністраторів. Також мають засоби підтримки віддалених користувачів (наприклад авторизація), і функції побудови віртуальних приватних мереж і т.д. До нього відносяться майже всі

наявні на ринку брандмауери. Функції фільтрації співвідносяться із OSI наступним чином (рисунок 1.1) .

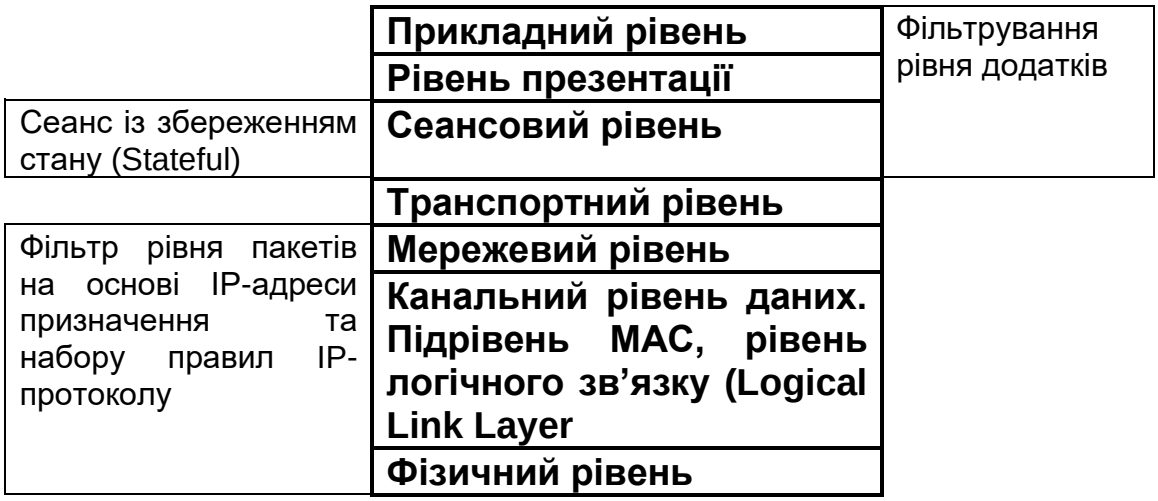

Рисунок 1.1 - Фільтрація пакетів в моделі OSI

1.2 Механізми аналізу трафіку

Міжмережевий екран (МЕ) для фільтрації пакетів використовує списки контролю доступу (access control lists ACL) на основі ідентифікатора протоколу верхнього рівня, IP-адреси джерела та призначення, TCP/UDP номерів портів джерела та призначення, а також напрямку передачі пакетів [9, 15-19].

При отриманні IP-датаграми брандмауер отримує заголовок пакета, а потім порівнює інформацію з заголовка пакета з правилами у списку контролю доступу, щоб вирішити, чи переслати або відкинути IP-датаграму. На рисунку 1.2 показано, як реалізується фільтрація пакетів на брандмауері.

Пристрій підтримує МЕ з фільтрацією пакетів і може фільтрувати описані нижче пакети.

Звичайні IP-пакети. МЕ перевіряє IP-адреси джерела та одержувача, номери портів джерела та одержувача, а також ідентифікатори протоколів IP-пакетів за списком ACL. Він пересилає пакети, дозволені списком ACL, і відкидає пакети, заборонені цим списком. Інформація, яку перевіряє

брандмауер, міститься в заголовку IP, TCP або UDP.

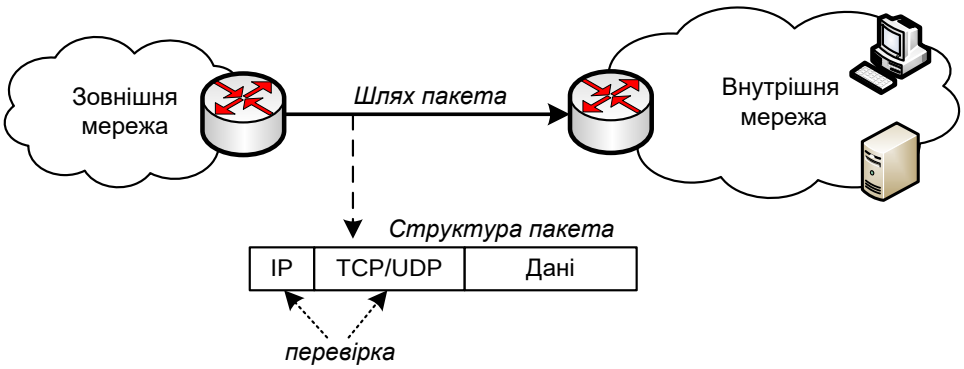

Рисунок 1.2 - Фільтрація пакетів

Фрагментація пакетів. МЕ може ідентифікувати типи пакетів, включаючи нефрагментовані пакети, пакети з початковим фрагментом і пакети без початкового фрагмента. При отриманні початкового фрагмента пакета брандмауер порівнює інформацію 3-го і 4-го рівнів початкового фрагмента зі списком контролю доступу. Якщо фрагмент дозволений, МЕ записує інформацію про цей фрагмент і створює таблицю відповідності для наступних фрагментів. Коли надходять наступні фрагменти, МЕ безпосередньо пересилає їх відповідно до таблиці відповідності.

Крім того, МЕ має метод за замовчуванням для обробки пакетів, які не відповідають списку правил. Метод за замовчуванням може бути встановлений користувачами.

Описані вище МЕ з фільтрацією пакетів є статичними і мають наступні проблеми:

– деякі політики безпеки не можуть бути налаштовані для багатоканальних протоколів прикладного рівня, таких як FTP і SIP;

– деякі атаки (TCP SYN) з транспортного та прикладного рівнів не можуть бути виявлені;

– атаки ICMP не можна запобігти, оскільки несправжні пакети помилок ICMP не можуть бути ідентифіковані;

– першим пакетом TCP-з'єднання повинен бути SYN-пакет. Якщо перший пакет TCP-з'єднання не є SYN-пакетом, пакет відкидається. Коли пристрій брандмауера вперше підключається до мережі, всі не перші пакети існуючих TCP-з'єднань відкидаються, якщо вони проходять через новий брандмауер, а TCP-з'єднання розриваються.

Для вирішення попередніх проблем вводиться фільтр пакетів, специфічний для конкретної програми (Application specific packet filter -ASPF), МЕ з підтримкою стану (Stateful), який вирішує ці проблеми. ASPF може виявляти атаки, пов'язані з наступними протоколами:

– протоколи прикладного рівня, включаючи протокол FTP і протокол HTTP, протокол ініціювання сеансу (SIP) і протокол потокового передавання в реальному часі (RTSP);

– протоколи транспортного рівня, включаючи TCP і UDP.

1.3 Аналіз роботи типового міжмережевого екрану

Сучасні мережні атаки часто використовують кілька складних методів для проникнення в корпоративні мережі. Щоб уникнути виявлення системами запобігання вторгнень, ці атаки часто кодуються за допомогою складних алгоритмів. Після одержання контролю зловмисник спробує завантажити й установити шкідливе ПЗ на "захоплений" пристрій. У багатьох випадках використовуються шкідливі програми новітніх версій, які не можуть виявити традиційні антивірусні засоби. Більше того, володіючи істотними трудовими й грошовими ресурсами, розроблювачі шкідливих рішень підлаштовуються під присутні на ринку розв'язку по захисту мереж, принцип їх роботи й аналізу трафіка. Наприклад, сучасні атаки використовують SSL-Шифрування, щоб сховати завантаження шкідливого ПЗ або навіть замаскувати трафік командного керування, переданого атакуючою особою. Або ж, знаючи про обмеженість у можливостях міжмережевих екранів попереднього покоління

обробляти великі пакети в істотній кількості, маскують шкідливий код у об’ємному трафіку.

Продукти Dell Sonicwall від молодших до старших моделей використовують запатентовану технологію "однопрохідного двигуна" з малим часом затримки Reassembly-Free Packet Inspection, який перевіряє кожний байт кожного пакета, зберігаючи при цьому високу швидкість передачі й продуктивність. Продукти Sonikwall для захисту мережі від внутрішніх і зовнішніх атак, що використовують уразливості застосувань, сканують увесь трафік, незалежно від розміру файлів, порту або протоколу. Можливості візуалізації й контролю використовуються для вивчення кожного пакета, щоб визначити які застосування використовуються, і хто їх використовує. Принцип роботи технології RFDI - розбивка вхідного пакета на сегменти й паралельне, одночасне сканування кожного сегмента. Дозволяє це реалізувати багатоядерна архітектура процесорів. Природно, що технологія RFPDI інтегрована із платформою самого пристрою й дозволяє оптимізувати керування деталізованими політиками міжмережевого екрана, як через інтерфейс міжмережевого екрана, так і через систему Dell Sonickwall Global Managment System. На рисунку 1.3 показано принцип сканування сканування без повторного формування пакету, без використання проксі і без обмежень по обсягу вмісту.

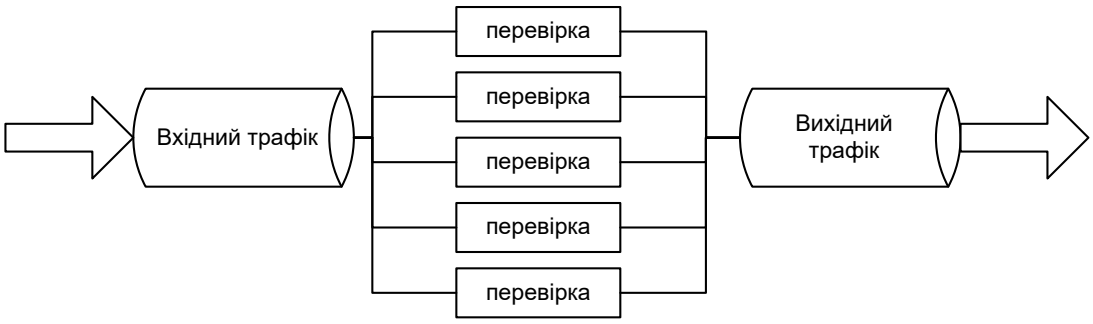

Рисунок 1.3 - Сканування без повторного формування пакету

Розшифрування й перевірка SSL-Трафіка є одним з головних і обов'язковим елементом забезпечення більш глибокого рівня мережної безпеки

в мережах нового покоління. Аналітики стверджують, що практично 35 відсотків корпоративного мережного трафіка шифрується за допомогою SSL. Таким чином, організації, які не мають відповідного встаткування й не перевіряють Ssl-Трафік, фактично залишають без перегляду одну третину трафіка в мережі, що привело до підвищеної уваги зловмисників саме до SSL, тому що гарантує стовідсотковий успіх атак таких організацій. Із цього випливає, що організація повинна мати можливість перевіряти весь трафік, і на будь-якому порту, незалежно - із шифруванням SSL він чи ні. Відповідну можливість надають міжмережеві екрани Dell Sonicwall, перевіряючи шифрований SSL і нешифрований трафік на кожному порту.

Крім того, міжмережеві екрани Dell Sonicwall використовують фірмовий ресурс мережі ідентифікації атак і моніторингу мережі Dell Sonicwall Global Response Intelligent Defense (GRID). Мережа являє собою більш 1 мільйона розкиданих по всій планеті, як їх називає виробник, "сенсорів". Ці сенсори збирають дані про підозрілу мережеву активність, шкідливе ПЗ, атаки і відправляють в умовний "Центр", де дані аналізуються на предмет реальності погрози, випускаються відновлення баз і сигнатур, після чого розсилаються всім пристроям Sonicwall на планеті.

Окремим питанням є відновлення баз і засобів мережного захисту від шкідливого ПЗ. Нам логічно зрозуміло, що якщо зловмисник випускає засіб, який використовує уразливість у якому-небудь ПЗ, то в зловмисника є карт-бланш на "збір урожаю" на цій уразливості доти, поки:

а) виробник уразливого ПЗ не зробить оновлення-патч і/або виробник захисної системи не обновить бази й засоби, що перешкоджають використанню уразливості;

б) забезпечуючі захист засоби не одержать відновлення.

У цілому, NGFW від Dell Sonicwall забезпечує інтеграцію засобів мережної безпеки, веб-захисту й безпеки електронної пошти. Це багаторівневий захист від вірусів, хробаків, троянів, шпигунського ПЗ і вторгнень. Веб-захист пристроїв забезпечує зручне керування блокуванням, що не відповідають

вимогам сайтів і контроль використання миттєвого обміну повідомленнями й пірінгових застосувань.

Рішення компанії Dell Sonicwall дозволяють розширити контроль над невиробничими застосуваннями, такими як онлайн-торгівля, обмін миттєвими повідомленнями/чатами, пірінговий обмін та потокова передача відео, шляхом їх аналізу, керування та візуалізації. Dell Sonicwall також надає рішення для захисту електронної пошти, що забезпечують захист від спаму та фішинг-атак, з метою захисту співробітників від отримання шахрайських та неправомочних електронних повідомлень. Інтелектуальні рішення Dell Sonicwall спрощують та знижують вартість централізованого керування локальними, віддаленими та мобільними мережними службами, забезпечуючи захист ключової інформації та комунікаційних ресурсів.

Міжмережеві екрани Dell Sonicwall надають можливість використовувати 3G/4 G-Зв'язок, що за невеликі гроші забезпечує організації принципово новий рівень резервування Інтернет-Доступу по альтернативному, виділеній лінії, фізичному способу передачі даних через бездротовий зв'язок, що в 00-х роках концептуально було доступне тільки компаніям з великими IT-бюджетами за рахунок організації каналів супутниковому зв'язку. Крім до цього, міжмережеві екрани Dell Sonicwall дозволяють мати декілька одночасно підключених WAN з'єднань і не тільки перекидати трафік з каналу на канал у випадку відмови робочого з'єднання, але й використовувати канали одночасно, балансуючи реальний трафік між ними на рівні застосувань, відправляючи, наприклад, критичний трафік на більш надійні й дорогі канали, а другорядний - на менш надійні й більш дешеві.

Для забезпечення контролю над застосуваннями Sonicwall використовує функцію Application Intelligence and Control, яка надає деталізований контроль і візуалізацію застосувань у режимі реального часу, забезпечуючи пріоритезацію пропускної здатності. Ця функція використовує запатентовану технологію Reassembly-Free Deep Packet Inspection (RFDPI) для розпізнавання і контролю застосувань незалежно від порту або протоколу. В даний момент база сигнатур розпізнає понад 4600 застосувань і мільйони видів шкідливого програмного

забезпечення, і постійно розширюється, забезпечуючи розподіл пропускної здатності та блокування доступу до веб-сайтів. Функція Application Flow Monitor дозволяє спостерігати в реальному часі за використанням застосувань, надаючи інформацію про вхідну й вихідну пропускну здатність, активні підключення до веб-сайтів та діяльність користувачів.

Технологія Dell Sonicwall Clean VPN призначена для забезпечення безпечного доступу й взаємодії з віддаленими користувачами при з'єднаннях Ipsec і SSL VPN. Сам VPN доступ захищається через аутентифікацію, шифрування даних і настроювання доступу. При цьому перевіряється одночасно як вхідний, так і вихідний VPN трафік. Перед потраплянням трафіка в мережу усередині "периметра", увесь VPN трафік дешифрується й очищається. На додаток, за допомогою функцій Application Intelligence and Control є можливість візуалізації трафіка в VPN тунелях і застосування політик по контролю смуги пропускання, пріоретизуючи трафік потрібних застосувань і блокуючи (або обмежуючи) трафік непотрібних.

Список протоколів, аналізованих типовою системою запобігання вторгнення Dell:

- HTTP / HTTPS;
- TCP / ICMP / DNS;
- FTP / FTPS;
- Telnet / SNMP;
- SMTP / IMAP / POP3;
- IPv4/IPv6/SSL;
- SIP / H.323;
- RTP / RPC;
- NETBIOS / SMB / SMB2;
- MySQL / MS-SQL.

1.4 Постановка задач кваліфікаційної роботи

Практичне значення роботи. Розроблення програмних засобів аналізу конфігурації маршрутизатора та правил фільтрації трафіку допомагає виконувати аудит мережевої інфраструктури та переконатися, що вона відповідає стандартам безпеки та правилам організації. Засоби аналізу конфігурації та правил фільтрації трафіку можуть служити інструментами для ефективного управління змінами в мережевій інфраструктурі [20].

Метою кваліфікаційної роботи є розроблення програмного модуля аналізу конфігурації маршрутизатора. Конфігурація маршрутизатора це інформація про налаштування мережевих інтерфейсів (включаючи IP-адреси, підмережі, маски підмереж, VLAN), маршрутних таблиць, правила мережевого екрану (правила фільтрації, перенаправлення портів, NAT) [21].

Для досягнення мети необхідно розв’язати такі задачі:

−проаналізувати функції маршрутизаторів та міжмережевих екранів;

−розробити об’єктну модель правил фільтрації;

−розробити алгоритми аналізу конфігурації;

−програмно реалізувати розроблені алгоритми з метою отримання правил фільтрації трафіку;

−здійснити тестування програмного засобу.

2 ЗАСОБИ АНАЛІЗУ КОНФІГУРАЦІЇ МАРШРУТИЗАТОРА

2.1 Алгоритми маршрутизації

Маршрутизація (Routing) - це процес визначення шляху передачі інформації між мережами. Роутер (або маршрутизатор) приймає рішення на основі IP-адреси отримувача пакета. Для пересилання пакета далі всі пристрої на шляху використовують IP-адресу отримувача. Щоб прийняти правильне рішення роутер повинен мати інформацію про напрямки та маршрути до віддалених мереж.

Спочатку маршрутизатори були реалізовані як спеціалізоване програмне забезпечення, яке обробляло вхідні IP-пакети за певним специфічним алгоритмом. Це програмне забезпечення працювало на комп'ютерах, які мали кілька мережевих інтерфейсів, що належали до різних мереж (які потребують маршрутизації). Згодом з'явилися спеціалізовані пристрої - маршрутизатори. Комп'ютери з програмним забезпеченням маршрутизатора називають програмними маршрутизаторами, а спеціальні пристрої - апаратними маршрутизаторами [15, 23].

У сучасних апаратних маршрутизаторах використовується спеціалізоване програмне забезпечення (так звана "прошивка") для створення таблиць маршрутизації. Для обробки IP-пакетів застосовується пристрій комутаційна матриця або інші апаратні технології комутації, які працюють з фільтрацією адрес в заголовку IP-пакета.

Існують два типи маршрутизації:

- статична маршрутизація, за якої маршрути встановлюються вручну адміністратором.

- динамічна маршрутизація, за якої маршрути обчислюються автоматично на основі протоколів динамічної маршрутизації. А саме OSPF, EIGRP, RIP, IS-IS, BGP, HSRP та інші. Перелічені протоколи беруть до уваги топологію і стан каналів зв'язку, опитуючи інші маршрутизатори у мережі.

Статичні маршрути налаштовуються вручну, будь-які зміни в мережній топології вимагають втручання адміністратора для додавання або видалення статичних маршрутів залежно від змін. У великих мережах підтримка статичних таблиць маршрутизації вручну може бути дуже часо- та працезатратною для адміністратора. В невеликих мережах це завдання може бути простішим. Статична маршрутизація не має масштабованості, яку може забезпечити динамічна маршрутизація, оскільки статична маршрутизація потребує додаткового налаштування та втручання адміністратора. Однак, навіть у великих мережах часто використовують комбінацію статичної маршрутизації з протоколами динамічної маршрутизації для спеціальних випадків, оскільки статична маршрутизація є більш стабільною і вимагає менше апаратних ресурсів маршрутизатора для управління таблицею маршрутизації [24-26].

Динамічні маршрути встановлюються за іншим принципом. Після активації та налаштування динамічної маршрутизації за одним з протоколів адміністратором. Поточна інформація про маршрути поновлюється автоматично в процесі маршрутизації якщо отримано нову інформацію про маршрути з мережі. Маршрутизатори взаємодіють між собою, обмінюючи повідомленнями про зміни у топології мережі під час динамічної маршрутизації.

Протокол маршрутизації працює лише з пакетами, які відносяться до одного з маршрутизованих протоколів, на сьогодні тільки IP. Стандарти протоколів маршрутизації визначають формат пакетів, зокрема заголовки, і найважливішою інформацією для маршрутизації є адреса одержувача. Протоколи не підтримуючі маршрутизацію, можуть передаватися між окремими мережами при допомозі тунелів. Ці можливості, як правило, доступні у програмних маршрутизаторах та деяких моделях апаратних маршрутизаторів.

У мережі Інтернет маршрутизація ґрунтується на протоколах TCP/IP. Отже передавання даних здійснюється шляхом використання IP-пакетів, у яких заголовок містить IP-адресу відправника та отримувача. Кожен пакет пропускається через маршрутизатор, який виконує обробку згідно зі своєю таблицею маршрутизації. Ця таблиця містить інформацію про те, до якого

комп'ютера направити пакети з певним діапазоном адрес. Зокрема, всі пакети, що належать до конкретного діапазону, можуть бути спрямовані до іншого маршрутизатора, який відповідає за цей сегмент [27-35].

У деяких ситуаціях маршрутизатор може змінювати заголовок пакета, змінюючи адресу надсилаючого вузла та/або одержувача. Це особливо стосується взаємодії локальної мережі з глобальною мережею Інтернет, яка має власні адреси. У таких випадках локальна мережа може бути доступною зовні лише за допомогою однієї глобальної IP-адреси. Для того, щоб маршрутизатор міг правильно направляти пакети до відповідних одержувачів у локальній мережі з використанням однієї глобальної адреси, використовується таблиця NAT. В цій таблиці, крім IP-адрес, також вказуються порти котрі позначають окремі додатки, що відкривають з'єднання. Інформація про порти розміщується в заголовку сегмента TCP або UDP, який вкладається у поле даних IP-пакета. Цей механізм забезпечує унікальну ідентифікацію відправника та одержувача, навіть якщо під однією глобальною адресою знаходиться багато комп'ютерів локальної мережі. У таблиці 2.1 наведено приклад вмісту таблиці NAT.

Таблиця 2.1 – Таблиця NAT

Глобальний адрес	Локальний адрес
208.164.201.225:1445	192.168.1.15:1445
208.164.201.225:1446	192.168.1.26:1445

У сфері комп'ютерних мереж існує кілька методів розсилки пакетів, серед яких головні методи представлені на рисунку 2.1. Один з цих методів - anycast («відправка даних кому завгодно») - дозволяє пристрою надсилати дані до найближчого одержувача з певної групи. Цей метод реалізований, зокрема, у протоколі IPv6.

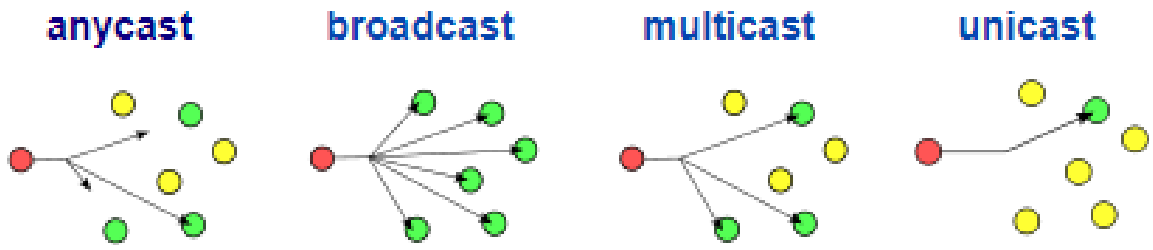

Рисунок 2.1 – Чотири механізми розсилки

У протоколі IP, механізм anycast реалізується шляхом опупублікування однакового маршруту в різних точках мережі за допомогою протоколу BGP (Border Gateway Protocol). Вибір конкретного маршруту в BGP залежить від критеріїв, а одним з них є AS-path - список номерів автономних систем, які пакет повинен пройти через них. BGP обирає маршрут AS-path, тобто з найкоротшим списком. Якщо отримано анонси маршрутів з декількох точок, обирається найкоротший шлях. Важливо зазначити, що найближчий вузол у механізмі anycast не обов'язково є географічно найближчим через особливості топології мережі або політики мережевих операторів. Механізм anycast широко використовується в Інтернеті для скорочення часу відгуку та застосовується для вирівнювання навантаження окремих кореневих DNS-серверів (балансування).

Широкомовлення (Broadcasting) є методом передачі даних в комп'ютерних мережах, при якому певний потік даних (або пакет при пакетній передачі) призначений для отримання всіма учасниками мережі [20, 35-40].

У протоколі TCP/IP широкомовна передача (broadcast) можливе лише в межах одного окремого сегмента мережі (рівнів L2, L3). Проте пакети даних можуть бути відправлені через границі сегмента, в якому відбувається широкомовна передача (наприклад, передача пакета на broadcast IP-адресу через маршрутизатор поза межами мережі). Навантаження на мережу в разі широкомовної передачі не відрізняється від механізму звичайної передачі одному отримувачі-адресату, оскільки пакети даних не дублюються (на відміну від передачі unicast).

Прикладом широкомовної передачі є використання протоколу ARP для визначення MAC-адреси конкретної IP-адреси. У цьому випадку відправляється

широкомовний пакет, який розповсюджується до всіх пристроїв, підключених до того самого L2-домену мережі. Прохідний пристрій з потрібною IP-адресою відповідає пакетом, в якому міститься відповідний MAC-адрес. Остання адреса в підмережі є широкомовною IP-адресою.

Мультимовлення (мультикаст), також відоме як багато-адресне мовлення або групова передача, є формою широкомовлення, при якій адресою призначення мережевого пакету є мультікастна група (від одного до багатьох пристроїв). Мультимовлення може використовуватись на канальному, мережевому і прикладному рівнях мережевого стеку для передачі даних одночасно кільком адресатам.

Деякі застосування, такі як дистанційне навчання, розсилка пошти, радіо, відео і відеоконференції, підтримують мультимовлення. У мережі з одним адресатом кожному отримувачеві встановлюється окреме з'єднання, навіть якщо вони споживають один і той же ресурс по загальному маршруту. У випадку багатоадресної розсилки джерело надсилає одну копію даних по загальному маршруту тим отримувачам, які підписалися на розсилку. Головна перевага такого підходу полягає в тому, що додавання нових користувачів не вимагає збільшення пропускної здатності мережі по спільному маршруту до кінцевих користувачів. В результаті зменшується навантаження на проміжне обладнання [10 ,11, 42].

У теорії комп'ютерних мереж, термін "unicast" або "одноадресна" передача даних означає передачу пакетів лише одному адресатові. Ця схема маршрутизації даних протилежна до широкомовної схеми маршрутизації.

2.2 Принципи фільтрації мережевих пакетів

NAT (Network Address Translation) маршрутизація - це процес перетворення IP-адрес мережі в пакетах даних під час їх передачі через мережу. NAT маршрутизація використовується для перетворення приватних IP-адрес на

публічні IP-адреси, що дозволяє приватним мережам (наприклад, домашнім мережам або внутрішнім мережам організацій) отримувати доступ до Інтернету за допомогою однієї або кількох публічних IP-адрес і зекономити публічні адреси [41, 43-45].

Під час NAT маршрутизації, мережний пристрій, такий як маршрутизатор або брандмауер, перетворює IP-адреси в заголовках пакетів, замінюючи приватну IP-адресу на публічну IP-адресу, яка може бути маршрутизована в Інтернеті. Крім того, NAT також використовує механізм переадресації портів (port forwarding), що дозволяє встановлювати зв'язок з конкретним пристроєм або послугою у приватній мережі через публічну IP-адресу.

Міжмережеві екрани і маршрутизатор взагалі (зокрема і пристрої Мікротік) мають складні алгоритми обробки пакетів. Типову процедуру обробки пакетів називають Packet Flow Chains [4].

Packet Flow Chains (послідовність обробки пакетів) - це поняття, яке використовується в контексті маршрутизаторів та комутаторів для опису шляху, яким пропускаються пакети через різні етапи обробки в пристрої. Кожен етап включає різні операції, такі як перевірка правил фільтрації, виконання маршрутизації, обробка заголовків пакетів тощо.

Packet Flow Chains описують логічну послідовність кроків, через які проходять пакети при проходженні крізь мережевий пристрій. Ці ланцюжки визначають, які операції виконуються на різних етапах обробки пакетів і в якому порядку.

Зазвичай, Packet Flow Chains складаються з таких етапів, як:

1. Вхідна ланка (Input Chain). Перша стадія, на якій пакети входять в пристрій, перевіряються на валідність, розпізнаються та визначаються їхній напрямок.
2. Ланка фільтрації (Filter Chain). На цьому етапі пакети проходять через правила фільтрації, де відбувається визначення, чи повинні вони бути дозволені чи заборонені для подальшої обробки.
3. Ланка маршрутизації (Routing Chain). Пакети, що успішно пройшли фільтрацію, надсилаються на маршрутизацію, де визначається

оптимальний шлях для пересилання пакетів до їхнього призначення.

4. Ланка обробки (Processing Chain). Після маршрутизації пакети проходять через різні операції обробки, такі як зміна заголовків, маніпуляції з поліпшенням якості обслуговування (QoS), шифрування, аутентифікація тощо.
5. Вихідна ланка (Output Chain) це останній етап, на якому пакети готуються до відправлення на вихідний інтерфейс, включаючи налаштування заголовків та можливу перетворення пакетів для сумісності з вихідним пристроєм або мережею.

На рисунку 2.2 наведено узагальнений алгоритми обробки пакетів.

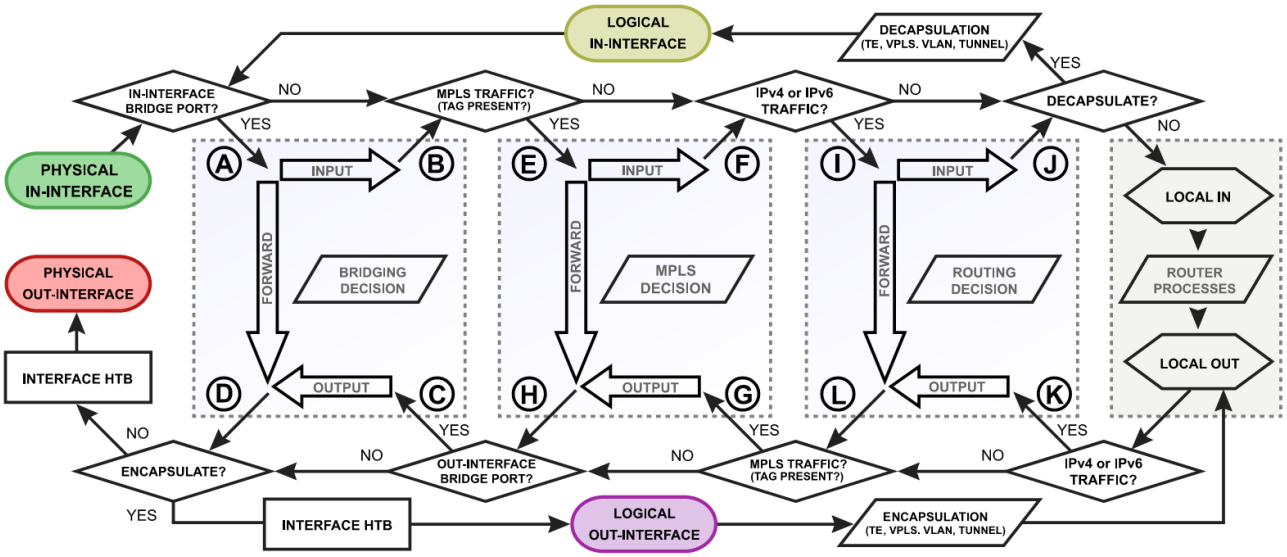

Рисунок 2.2 - Узагальнений алгоритми обробки пакетів в маршрутизаторі

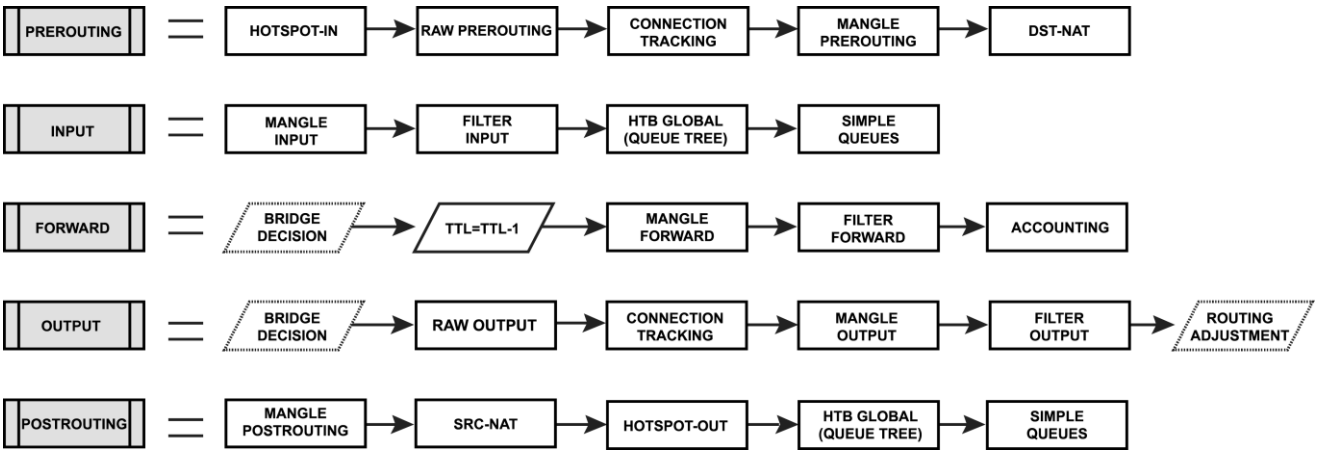

Рисунок 2.3 – Деталізація типових процедур

2.3 Об’єктна модель правил фільтрації

Схема правил маршрутизаторів різних класів відрізняється значною мірою. Спільною частиною, можна сказати, є структура "ЯКЩО" для визначення умови і "ДІЯ" для виконання дії над пакетом [10, 15, 41-43].

Структура правила для фільтрації трафіку в маршрутизаторі зазвичай містить наступні компоненти:

1. Умова (Condition): Це частина правила, яка визначає, які умови повинні бути виконані для застосування правила до конкретного пакету. Умови можуть включати такі параметри, як IP-адреси джерела та призначення, порти, протоколи і т.д.

2. Дія (Action): Це частина правила, яка визначає, що робити з пакетом, якщо він відповідає заданим умовам. Дії можуть включати блокування пакету, пересилання на інший інтерфейс, зміну атрибутів пакету і т.д.

3. Пріоритет (Priority): Це значення, яке вказує важливість правила у порівнянні з іншими правилами фільтрації. При виявленні пакету, який відповідає декільком правилам, виконується правило з найвищим пріоритетом.

4. Напрямок (Direction): Це вказівка, яка визначає, чи застосовується правило до вхідного трафіку (ініційованого зовнішніми джерелами) чи вихідного трафіку (ініційованого внутрішніми джерелами).

5. Джерело та призначення (Source and Destination): Ці параметри визначають джерело та призначення пакету, які можуть бути вказані у вигляді IP-адрес, підмереж, діапазонів портів тощо.

6. Інші атрибути (Other Attributes): В деяких випадках правила можуть містити інші атрибути, такі як часові обмеження, пріоритет обробки, маркування пакетів і т.д.

Загальна структура правила може варіюватися залежно від конкретної системи маршрутизації та її конфігурації.

У роботі була розроблена об'єктна модель для представлення правил фільтрації, яка показана на рисунку 2.5. Клас "FilterRule" виконує функцію узагальненого правила фільтрації. Класи "TP LINK FilterRule" і "Mikrotik FilterRule" представляють правила фільтрації з відповідними атрибутами та методами, що відповідають програмній моделі відповідних пристроїв.

На діаграмі класів (рисунок 2.5), представлена структура опису правила фільтрації.

Атрибути класу "FilterRule ":

– "In address", "Out address" - адреси відправника та отримувача;
– "In port", "Out port" - порти відправника та отримувача.

Методи класу " FilterRule ":

– "Add filter rule()" і "Delete filter rule()" - додавання та видалення правила;
– "Enable filter rule()" і "Disable filter rule()" - активація та деактивація правила.

Атрибути класу "TP LINK FilterRule ":

– " Filter Rule name" - назва правила;
– "Host Descr" - опис вузла;
– "Address Type" - тип адреси;
– "LAN IP_start", "LAN IP_stop" - діапазон IP-адрес;
– "Target descr" - опис цілі.

Методи класу "TP LINK FilterRule ":

– "Add_Host()" - додавання вузла;
– "Add_target()" - додавання цілі;
– "Set_schedule()" - задання розкладу для запуску правила;
– "Get_status()", "Set_status()" - отримання та зміна статусу.

Атрибути класу "Microtik Firewall FilterRule ":

– "Chain" - ланцюг;
– "Src_Address", "Dst_Address" - адреси відправника та одержувача;
– "Protocol" - протокол;

– "Src_Port", "Dst_Port" - порти відправника та одержувача;

– "Action" - дія з даним пакетом.

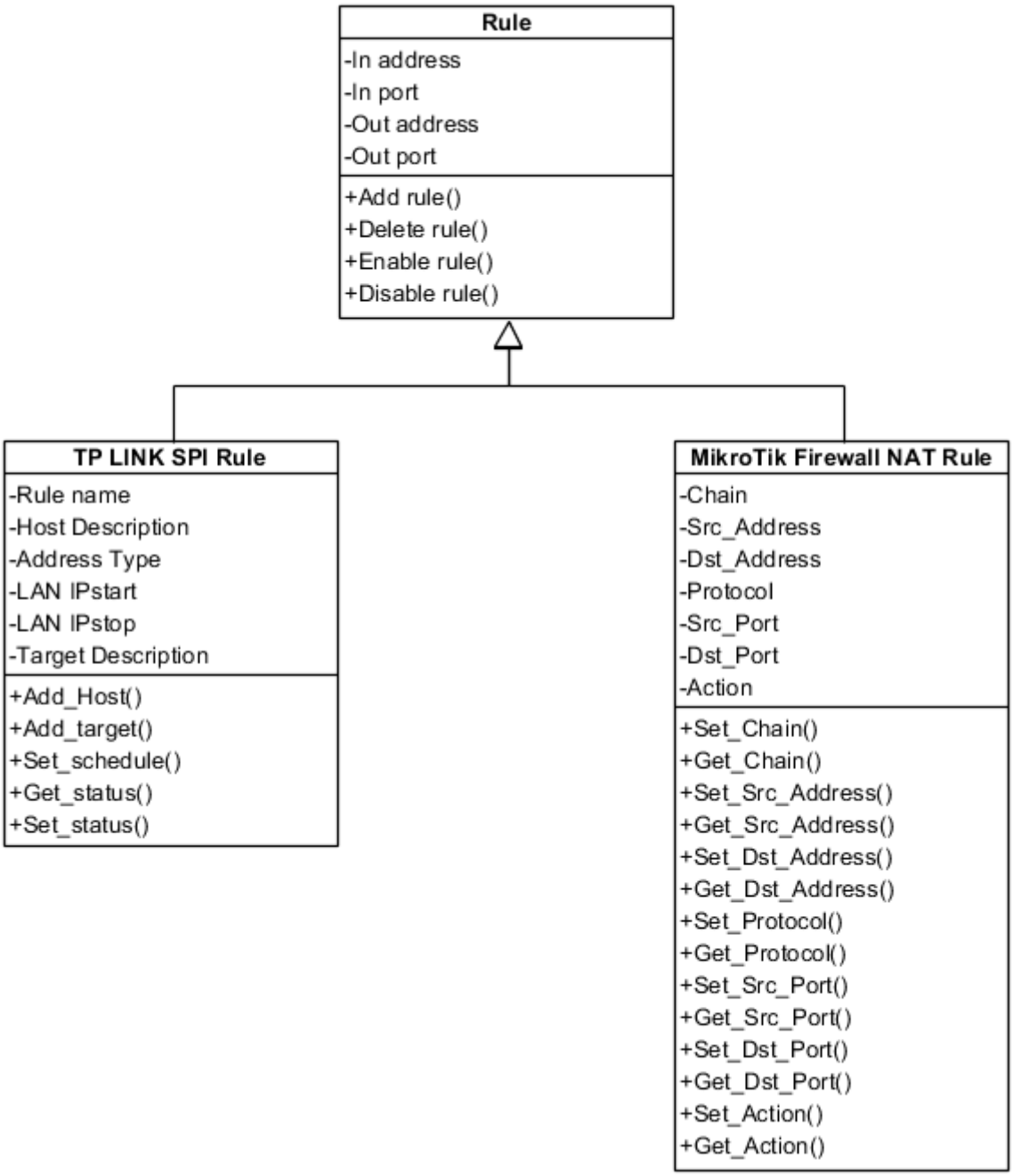

Рисунок 2.5 – Діаграма класів для створення правил фільтрації

Методи класу "Microtik Firewall FilterRule ":

– "Set_RChain()", "Get_RChain()" - встановлення та отримання значення ланцюжка;

– "Set_Source_Address()", "Get_Source_Address()", "Set_Dst_Address()", "Get_Dst_Address()" - встановлення та отримання значень адрес відправника та отримувача;

– "Set_Protocol()", "Get_Protocol()" - встановлення та отримання значення протоколу;

– "Set_Source_Port()", "Get_Source_Port()", "Set_Dst_Port()", "Get_Dst_Port()" - встановлення та отримання значень портів відправника та отримувача;

– "Set_Action()", "Get_Action()" - встановлення та отримання значення дії, що виконується над пакетом.

Розглянемо основні функції та призначення команд операційної системи RouterOS для формування конфігурації маршрутизатора у вигляді тексту.

Мова сценаріїв в операційній системі пристроїв Мікротік використовується для автоматизації рутинних налаштувань. Нижче перераховані основні етапи типового налаштування маршрутизатора:

- Блокувати всіх, що знаходяться у чорному списку.
- Фільтрувати корисні ICMP-пакети.
- Блокувати мережі Bogon.
- Блокувати DNS-запити на зовнішній інтерфейс.
- Застосовувати захист від брутфорсу для сервісу SSH.
- Застосовувати захист від сканера портів.
- Дозволяємо вже встановлені підключення та пов'язані з ними станом connection-state=established.
- Дозволяємо зовнішні підключення для власних потреб.
- Дозволяємо вже встановлені підключення та пов'язані з ними ланцюжком chain=forward та станом connection-state=established.

Операційна система MikroTik RouterOS має широкий набір команд, які використовуються для конфігурації, керування та моніторингу. Основні типи команд описано нижче.

1.Команди налаштування інтерфейсів, такі як Ethernet, Wi-Fi, VLAN та інші. Ці команди дозволяють створювати, змінювати та видаляти інтерфейси та встановлювати параметри, такі як IP-адреси, маски підмережі, шлюзи тощо.

`interface add`, `interface set`, `interface remove`, `interface print`.

2.Налаштування маршрутизації. RouterOS підтримує різні протоколи маршрутизації, такі як OSPF, RIP, BGP та інші. Команди дозволяють налаштовувати параметри протоколів маршрутизації, визначати маршрути, виконувати фільтрацію маршрутів та моніторити стан маршрутизації.

`routing ospf`, `routing rip`, `routing bgp` та `routing print`.

3.Налаштування файрволу: RouterOS має вбудовану функцію файрволу, яка дозволяє керувати трафіком в мережі. Ці команди дозволяють створювати правила файрволу, налаштовувати NAT (Network Address Translation), редагувати заголовки пакетів та інше.

`ip firewall filter`,
`ip firewall nat`, `ip firewall mangle`, `ip firewall print`.

4.Налаштування безпеки. Маршрутизатор має різні функції безпеки для захисту мережі та маршрутизатора. Ці команди дозволяють створювати та налаштовувати користувачів, встановлювати паролі, налаштовувати правила файрволу та забезпечувати безпеку мережі.

`user add`, `user set`, `password`, `firewall`

5.Моніторинг, діагностика в стан маршрутизатора. Ці команди дозволяють виконувати пінгування хостів, відстежувати шляхи пакетів, моніторити ресурси системи, такі як CPU, пам'ять, пропускну здатність тощо.

`ping, traceroute, interface monitor, system resource print`

Для того щоб отримати конфігурацію маршрутизатора в програмний додаток потрібно розглянути технології та засоби управління маршрутизаторами в також вбудовані операційні системи.

MikroTik RouterOS, операційна система, що працює на маршрутизаторах MikroTik, надає різні засоби та протоколи для управління маршрутизаторами. Основні засоби та протоколів управління

1. CLI (Command Line Interface): RouterOS надає текстовий інтерфейс командного рядка, який дозволяє адміністраторам взаємодіяти з маршрутизатором, виконуючи команди. CLI дозволяє налаштовувати маршрутизатор, створювати правила мережевого екрану, налаштовувати мережеві інтерфейси та інші параметри.

2. Web-інтерфейс: MikroTik також надає графічний інтерфейс користувача через веб-браузер. Веб-інтерфейс дозволяє адміністраторам налаштовувати маршрутизатор за допомогою графічного інтерфейсу.

3. API (Application Programming Interface): RouterOS має API, який дозволяє розробникам створювати свої власні програми або інтегрувати існуючі програми з маршрутизаторами MikroTik. API надає можливість зчитувати і записувати конфігурацію (двійкову), виконувати команди, отримувати статистику та інше.

4. SNMP (Simple Network Management Protocol): MikroTik підтримує SNMP, стандартний протокол управління мережевими пристроями. SNMP дозволяє зчитувати статус мережевих інтерфейсів, статистику, налаштовувати SNMP Trap, які повідомляють про події в мережі, та інше.

5. Winbox - це окрема програма для управління маршрутизаторами MikroTik, доступна для платформ Windows. Winbox надає зручний графічний інтерфейс, який дозволяє налаштовувати маршрутизатори та виконувати різні дії, такі як налаштування IP-адрес, мережевих інтерфейсів, VPN-з'єднань, мережевої безпеки та багато іншого.

Управління маршрутизаторами можна здійснювати через протокол SSH. Це безпечний засіб для конфігурації та керування мережевими пристроями. Протокол SSH (Secure Shell) є криптографічним протоколом, який забезпечує безпечне з'єднання та комунікацію між двома вузлами, такими як комп'ютери, сервери або маршрутизатори. Основні аспекти управління маршрутизатором через протокол SSH:

1. Підключення до маршрутизатора передбачає автентифікацію з використанням імені користувача та пароля або, що більш безпечно, з використанням ключа SSH. Використання ключа SSH дозволяє уникнути передачі пароля через мережу, забезпечуючи більшу безпеку.

2. SSH забезпечує криптографічне шифрування всієї комунікації між клієнтом і сервером. Це означає, що всі дані захищені від перехоплення та зловживання.

3. Після успішного підключення до маршрутизатора можна виконувати команди на маршрутизаторі з клієнтської сторони. Це дозволяє вам змінювати конфігурацію маршрутизатора, створювати правила файрволу, налаштовувати мережеві інтерфейси та виконувати інші дії, які доступні через командний рядок або API.

4. Протокол дозволяє дистанційне управління маршрутизатором MikroTik з будь-якого комп'ютера або пристрою. Це дає змогу адміністраторам віддалено налаштовувати, управляти безпосередньо з місця розташування.

У третьому розділі ми використаємо функцію виконувати команди на маршрутизаторі з клієнтської сторони через SSH.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБПЕЗПЕЧЕННЯ

3.1 Інструментальні засоби розроблення мережевого додатку

Secure Shell (SSH) - це протокол для безпечного мережевого зв'язку, розроблений як відносно простий і недорогий у впровадженні. Початкова версія SSH1 була зосереджена на забезпеченні безпечного віддаленого входу в систему на заміну TELNET та інших схем віддаленого входу в систему, які не забезпечували ніякої безпеки. SSH також надає більш загальні можливості клієнт/сервер і може використовуватися для таких мережевих функцій, як передача файлів і електронна пошта. Нова версія, SSH2, виправляє ряд недоліків безпеки в оригінальній схемі. SSH2 задокументовано як запропонований стандарт в IETF RFCs 4250 - 4256.

Клієнтські та серверні програми SSH широко доступні для більшості операційних систем. Він став методом вибору для віддаленого входу в систему і X-тунелювання і швидко стає одним з найбільш поширених додатків для технології шифрування за межами вбудованих систем.

SSH складається з трьох протоколів, які зазвичай працюють поверх TCP (рисунок 3.1):

Протокол транспортного рівня Transport Layer Protocol:: Забезпечує автентифікацію сервера, конфіденційність і цілісність даних з прямим шифруванням (тобто, якщо ключ скомпрометований під час одного сеансу, це не впливає на безпеку попередніх сеансів). Транспортний рівень може додатково забезпечувати стиснення.

Протокол автентифікації користувача User Authentication Protocol:: Аутентифікує користувача на сервері.

Протокол з'єднання Connection Protocol:: Мультиплексує декілька логічних каналів зв'язку над одним базовим SSH-з'єднанням.

Рисунок 3.1 - Стек протоколів SSH

Розглянемо протокол транспортного рівня.

Автентифікація сервера відбувається на транспортному рівні, на основі пари відкритих/закритих ключів сервера. Сервер може мати кілька хост-ключів, що використовують різні асиметричні алгоритми шифрування. Кілька хостів можуть використовувати один і той самий ключ хоста. У будь-якому випадку, ключ хоста сервера використовується під час обміну ключами для автентифікації особи хоста. Щоб це стало можливим, клієнт повинен апріорі знати відкритий ключ хоста сервера. RFC 4251 диктує дві альтернативні моделі довіри, які можуть бути використані:

1. Клієнт має локальну базу даних, яка пов'язує кожне ім'я хоста (введене користувачем) з відповідним відкритим ключем хоста. Цей метод не вимагає централізованого адміністрування інфраструктури та координації з боку третьої сторони. Недоліком є те, що база даних зв'язків між іменами та ключами може стати обтяжливою в обслуговуванні.

2. Асоціація "ім'я-ключ" хоста сертифікована довіреним центром сертифікації (ЦС, certification authority). Клієнт знає лише кореневий ключ ЦС і може перевірити дійсність усіх ключів хостів, сертифікованих визнаними ЦС.

Ця альтернатива полегшує проблему обслуговування, оскільки в ідеалі на клієнті потрібно надійно зберігати лише один ключ ЦС. З іншого боку, кожен ключ хоста повинен бути належним чином сертифікований центральним органом, перш ніж авторизація стане можливою.

ОБМІН ПАКЕТАМИ Рисунок 16.9 ілюструє послідовність подій у протоколі транспортного рівня SSH. Спочатку клієнт встановлює TCP-з'єднання з сервером. Це робиться за допомогою протоколу TCP і не є частиною протоколу транспортного рівня. Після встановлення з'єднання клієнт і сервер обмінюються даними, які називаються пакетами, в полі даних TCP-сегмента. Кожен пакет має наступний формат (рисунок 3.2).

- Довжина пакета: Довжина пакета в байтах, не включаючи поля довжини пакета і MAC-адреси.

- Довжина заповнення Padding: Довжина довільного поля заповнення.

- Корисний вміст Payload: Корисний вміст пакета. До узгодження алгоритму це поле не стискається. Якщо стиснення узгоджено, то в наступних пакетах це поле стискається.

- Випадкове заповнення Padding: Після узгодження алгоритму шифрування це поле додається. Воно містить випадкові байти заповнення таким чином, щоб загальна довжина пакета (за винятком поля MAC) була кратною розміру блоку шифрування, або 8 байт для потокового шифрування.

- Код автентифікації повідомлення (КАП): Якщо автентифікація повідомлення була узгоджена, це поле містить значення КАП. Значення КАП обчислюється для всього пакета плюс порядковий номер, за винятком поля MAC. Порядковий номер - це неявна 32-розрядна послідовність пакетів, яка ініціалізується значенням нуль для першого пакета і збільшується для кожного наступного пакета. Номер послідовності не включається до пакету, що надсилається через TCP-з'єднання.

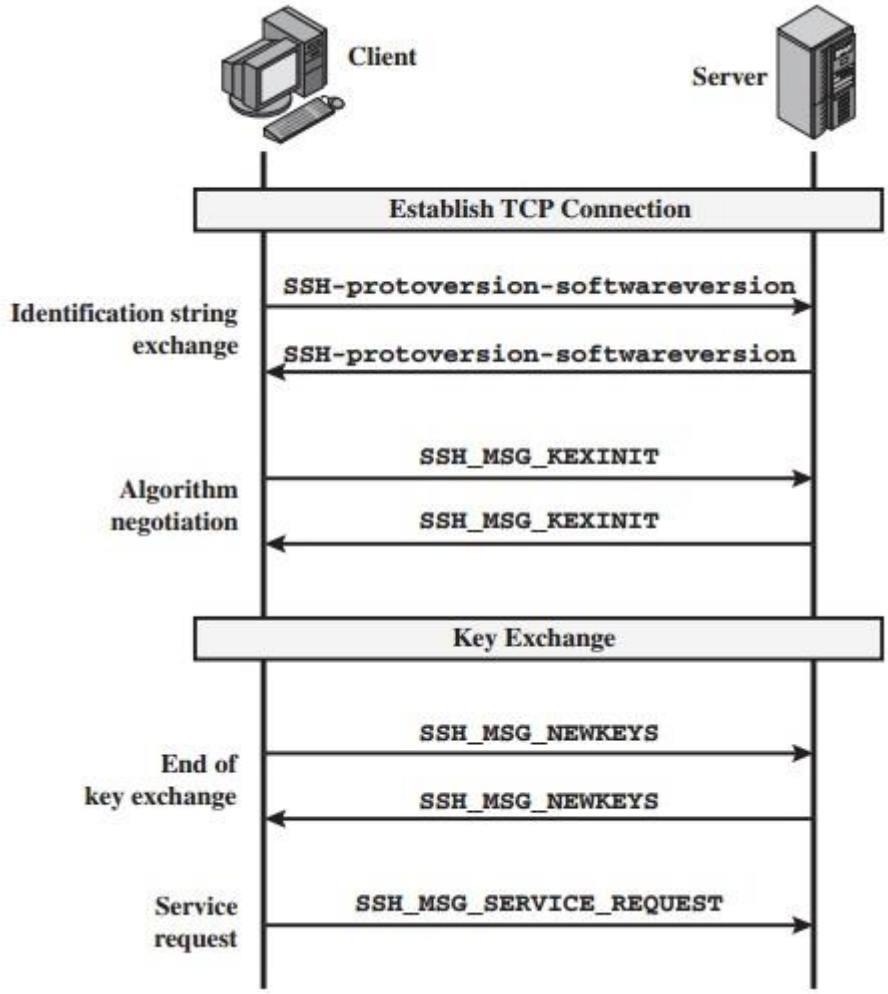

Рисунок 3.2 - Обмін пакетами протоколу транспортного рівня SSH

Опишемо методи автентифікації. Сервер може вимагати один або декілька з наступних методів автентифікації.

- публічний ключ: Деталі цього методу залежать від обраного алгоритму відкритого ключа. По суті, клієнт надсилає серверу повідомлення, яке містить відкритий ключ клієнта, а також підписує повідомлення закритим ключем клієнта. Коли сервер отримує це повідомлення, він перевіряє, чи є наданий ключ прийнятним для автентифікації, і якщо так, то перевіряє правильність підпису.

- password: Клієнт надсилає повідомлення, що містить пароль у відкритому вигляді, який захищено шифруванням протоколом транспортного рівня.

- на основі хоста: Автентифікація виконується на хості клієнта, а не на самому клієнті. Таким чином, хост, який підтримує декілька клієнтів, забезпечує автентифікацію для всіх своїх клієнтів. Цей метод працює, коли клієнт надсилає підпис, створений за допомогою приватного ключа хоста клієнта. Таким чином, замість того, щоб безпосередньо перевіряти особу користувача, SSH-сервер перевіряє особу клієнтського хоста, а потім вірить хосту, коли той стверджує, що користувач вже пройшов автентифікацію на стороні клієнта.

Paramiko - це бібліотека Python, яка встановлює з'єднання з віддаленим пристроєм через SSh. Paramiko використовує SSH2 як заміну SSL для створення безпечного з'єднання між двома пристроями.

Тестове середовище для відлагодження розробленого програмного забезпечення складається з ноутбука Dell та маршрутизатора Mikrotik RB951Ui-2HnD (рисунок 3.3).

Рисунок 3.3 – Тестове обладнання Mikrotik RB951Ui-2HnD

MikroTik RB951Ui-2HnD - це Wi-Fi роутер на 2,4 ГГц з достатньою продуктивністю і канальною швидкістю до 300 Мбіт/с. Маршрутизатор працює в бездротових стандартах IEEE 802.11b/g/n. Усередині пристрою встановлено потужний процесор із частотою 600 МГц, 128 МБ оперативної пам'яті та дві антени по 2,5 дБі. Максимальна вихідна потужність бездротового модуля становить 30 дБм. Маршрутизатор має п'ять Ethernet портів 10/100 Мбіт/с і

один USB порт для під'єднання зовнішніх накопичувачів або 3G модемів. Через п'ятий мережевий порт можна живити інший мережевий пристрій за технологією POE. Детальні характеристики [24] наведені в таблиці 3.1

Таблиця 3.1 - Характеристики MikroTik RB951Ui-2HnD

Характеристика	Значення
Система	
Процесор	Atheros AR9344 600 МГц
RAM	128 MB DDR SDRAM
Flash	64 MB
Інтерфейси	5×10/100 Мбіт/с LAN, (5-й порт з PoE), 1×USB 2.0
ОС	MikroTik RouterOS Level4
Точка доступу	
Стандарти	IEEE 802.11 b/g/n
Канальна швидкість	300 Мбіт/с
Додаткові функції	
Керування пристроєм	winbox, telnet, WEB інтерфейс

3.2 Розроблення об’єктної моделі та UML діаграм

Відповідно до поставленої мети потрібно розробити функції завантаження і представлення користувачу наступної інформації:

– налаштування мережевих інтерфейсів а саме, IP-адреси інтерфейсів, підмережі, списки груп VLAN);

– таблиць маршрутів;

– правил мережевого екрану а саме правил фільтрації, перенаправлення портів, NAT.

Створенням UML діаграми в форматі тексту, яка відображатиме основні використовувані випадки (use case) для функції парсингу правил фільтрації трафіка маршрутизатора.

```
@startuml
left to right direction

actor User as User
rectangle "Парсер правил\nфільтрації трафіка" as Parser {
    usecase "Завантаження правил\nфільтрації" as UC1
    usecase "Аналіз та розбір\nправил" as UC2
    usecase "Застосування правил\nдо трафіку" as UC3
    usecase "Виведення результату\nфільтрації" as UC4
}
```

Опис зв’язків між сутностями.

```
User --> UC1: Ввід правил
User --> UC4: Виведення результату

UC1 --> UC2: Аналіз та розбір
UC2 --> UC3: Застосування
UC3 --> UC4: Фільтрація
@enduml
```

Ця use case UML діаграма (рисунок 3.4) демонструє основні кроки, які відбуваються в функції парсингу правил фільтрації трафіка маршрутизатора. Користувач вводить правила, які потім передаються до парсера. Парсер аналізує та розбирає правила, після чого застосовує їх до трафіку. Результати фільтрації виводяться для перегляду користувачем. Діаграма виглядає наступним чином

Узагальнена структура програмного засобу наведено на кресленні КР.КІ.9675963.00.00.001.С1. Основні блоки: консольний інтерфейс, зчитування конфігурації з пристрою, запис конфігурації у пристрій, SSH інтерфейс маршрутизатора, редагування правил фільтрації, список правил фільтрації, перевірка коректності, візуалізація правил.

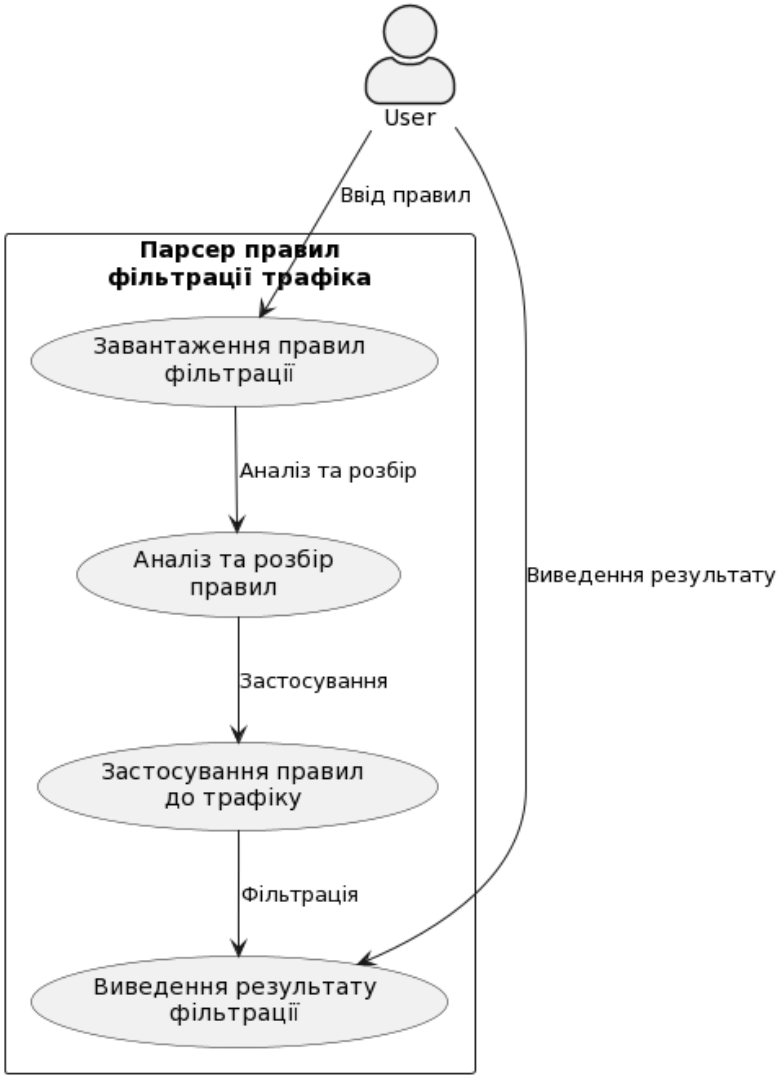

Рисунок 3.4 – Функція аналізу (парсингу) правил фільтрації трафіка

Основні блоки програмного засобу аналізу конфігурації маршрутизатора можуть бути описані наступним чином:

1. Консольний інтерфейс відповідає за взаємодію з користувачем через командний рядок. Він може надавати команди та опції для виконання різних дій, таких як зчитування конфігурації, запис конфігурації, редагування правил фільтрації і т.д..

2. Зчитування конфігурації з пристрою відповідає за зчитування поточної конфігурації з маршрутизатора або іншого пристрою мережі. Він використовує відповідні протоколи наприклад, SSH, Telnet .

3. Запис конфігурації у пристрій дозволяє користувачу внести зміни в конфігурацію маршрутизатора та зберегти їх на пристрої.

4. SSH інтерфейс маршрутизатора забезпечує з'єднання та аутентифікацію з маршрутизатором через протокол SSH (Secure Shell). Він дозволяє взаємодіяти з маршрутизатором за допомогою команд та операцій, які надаються протоколом SSH.

5. Редагування правил фільтрації дозволяє користувачу виконувати додавання, видалення або зміни правил фільтрації, які використовуються для керування трафіком у мережі.

6. Список правил фільтрації відображає поточний список правил фільтрації, які застосовуються на маршрутизаторі. Він може відображати деталі кожного правила, такі як джерело, призначення, протокол, порти і дії, які виконуються над пакетами, що відповідають правилу.

7. Блок Перевірка коректності виконує перевірку коректності правил фільтрації на маршрутизаторі. Він може виявляти потенційні проблеми, конфлікти або помилки у налаштуваннях, які можуть впливати на роботу мережі.

8. Візуалізація правил фільтрації надає візуальне представлення правил фільтрації у вигляді таблиць .

Узагальнений алгоритм роботи програмного засобу наведено на кресленні КР.КІ.9675963.00.00.002.С1. Інтерфейс програмного засобу керування маршрутизатором має три пункти меню: Мережеві інтерфейси, Марштрутні таблиці, Правила фільтрації.

1. Пункт «Мережеві інтерфейси» забезпечує:

а. Перегляд інформації про наявні мережеві інтерфейси, такі як назва інтерфейсу, стан (активний/неактивний), IP-адреса, підмережа, маска, швидкість передачі даних тощо.

б. Налаштування параметрів мережевих інтерфейсів, такі як зміна IP-адреси, підмережі, маски, налаштування швидкості передачі даних, встановлення/зняття фільтрів трафіку та ін.

2. Пункт «Маршрутні таблиці» забезпечує:

а. Перегляд інформації про поточні маршрути, включаючи мережеву адресу призначення, IP-адресу наступного маршрутизатора, інтерфейс для вихідного з'єднання, метрику маршруту тощо.

б. Додавання, видалення або зміна маршрутів.

в. Виконання операцій з маршрутизацією, таких як перенаправлення пакетів, встановлення пріоритетів маршрутизації та ін.

3. Пункт «Правила фільтрації» забезпечує:

а. Перегляд інформації про поточні правила фільтрації трафіку, включаючи джерело, призначення, протокол, порти, дію (дозволити/заборонити) тощо.

б. Додавання, видалення або зміна правил фільтрації.

в. Налаштування екрану, контролю доступу.

Специфікація функцій класу для консольного інтерфейсу на мові Python:

```
class ConsoleInterface:
    def __init__(self):
        # Ініціалізація консольного інтерфейсу
    def display_menu(self):
        # Відображення головного меню програми
    def handle_menu_selection(self, choice):
        # Обробка вибраного пункту меню
    def display_network_interfaces(self):
        # Відображення інформації про мережеві інтерфейси
    def configure_network_interface(self, interface_name):
        # Налаштування параметрів мережевого інтерфейсу
    def display_routing_tables(self):
        # Відображення інформації про маршрутні таблиці
    def configure_routing_table(self, route):
        # Налаштування маршрутних таблиць
```

Функцій для консольного інтерфейсу для правил фільтрації.

```
    def display_filter_rules(self):
        # Відображення інформації про правила фільтрації
    def configure_filter_rule(self, rule):
        # Налаштування правил фільтрації
    def run(self):
        # Запуск консольного інтерфейсу
```

Зобразимо дану специфікацію на UML діаграмі (рисунок 3.5).

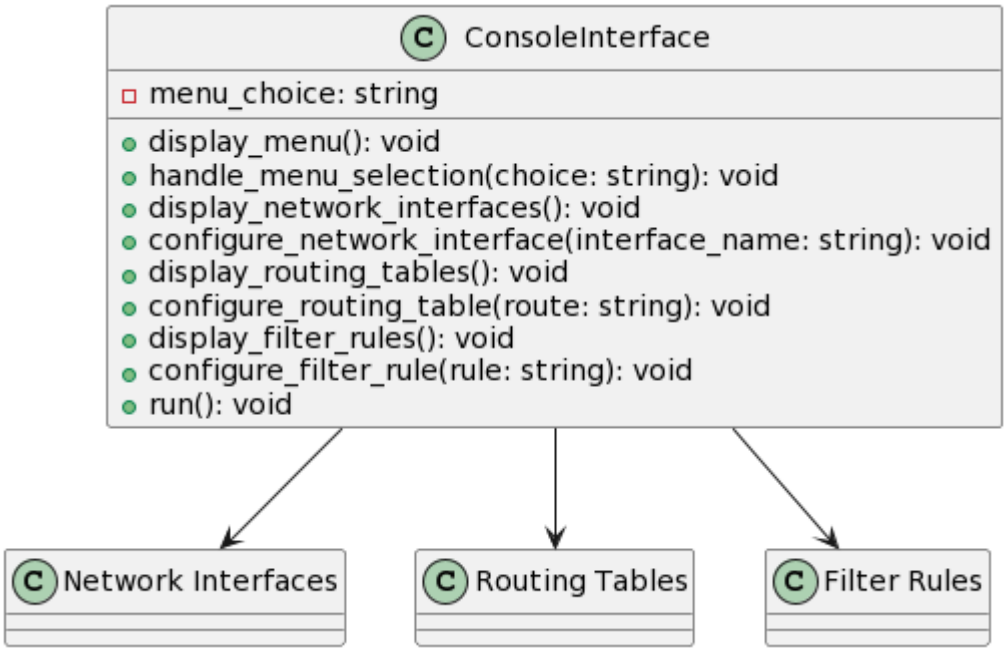

Рисунок 3.5 - Специфікація класу для консольного інтерфейсу

3.3 Реалізація та тестування програмного забезпечення

Спочатку необхідно підключити бібліотеку paramiko через інструкцію

```
pip install paramiko
```

Встановити параметри авторизації: IP-адресу пристрою, 'ім’я-користувача', 'пароль'.

```
import paramiko
# Змінні для SSH-підключення
host = '192.168.0.1'
port = 22
username = 'ім’я-користувача'
password = *****
# Функція для виконання команди SSH
def execute_command(ssh, command):
    stdin, stdout, stderr = ssh.exec_command(command)
    output = stdout.read().decode('utf-8')
    return output
# Встановлення SSH-з'єднання
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh.connect(host, port, username, password)
```

Функція аналізу маршрутних таблиць в MikroTik RouterOS, використовуючи Python та бібліотеку paramiko для SSH-підключення до пристрою.

```
# Отримання вихідної інформації про маршрути
route_info = execute_command(ssh, '/ip route print')

# Обробка та виведення результатів
routes = route_info.split('\n')
for route in routes:
    if route.startswith(' '):  # Пропуск заголовків та пустих
        continue
    columns = route.split()
    # Перевірка наявності необхідних колонок
    if len(columns) >= 4:
        destination = columns[0]
        gateway = columns[1]
        interface = columns[4]
        print(f'Destination: {destination}, Gateway: {gateway},
Interface: {interface}')

# Закриття SSH-з'єднання
ssh.close()
```

Цей код виконує SSH-підключення до пристрою, отримує вихідну інформацію про маршрути за допомогою команди /ip route print і виводить отримані дані про маршрути на консоль. Повний текст програмного модуля, включаючи дану та наступні функції наведений в додатку А.

Реалізуємо функцію аналізу інформації про інтерфейси.

```
def analyze_router_interfaces(ip, username, password):
    # Підключення до пристрою
    client = paramiko.SSHClient()
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    client.connect(ip, username=username, password=password)
    # Виконання команди на пристрої
    stdin, stdout, stderr = client.exec_command("/interface print
detail")
```

Перетворюємо отриманий текст в структурований вигляд довідника Python з іменованими змінними.

```
    # Отримання результатів виконання команди
    output = stdout.read().decode()
    # Розбиття результатів на рядки
    lines = output.split("\n")
```

```
# Аналіз інформації про інтерфейси
interfaces = []
for line in lines:
    if line.startswith("Flags"):
        continue
    elif line.startswith("  "):
        # Розбиття рядка на поля
        fields = line.strip().split()
        interface = {
            "name": fields[0],
            "type": fields[1],
            "mtu": fields[2],
            "mac_address": fields[3],
            "status": fields[4],
        }
        interfaces.append(interface)
```

Виведення результатів у вигляді іменованих змінних

```
for interface in interfaces:
    print("Name:", interface["name"])
    print("Type:", interface["type"])
    print("MTU:", interface["mtu"])
    print("MAC Address:", interface["mac_address"])
    print("Status:", interface["status"])
    print()

# Закриття з'єднання
client.close()
```

Функція виконує SSH-підключення до пристрою MikroTik RouterOS, виконує команду /interface print detail і аналізує результати, виводячи інформацію про інтерфейси, такі як імена, типи, MTU, MAC-адреси та статуси.

Тестування функції аналізу маршрутних таблиць показано на рисунку 3.6

```
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADS  0.0.0.0/0                          5.58.64.1                 1
 1 ADC  5.58.64.0/19       5.58.78.48      ether1                    0
 2 ADC  192.168.0.0/24     192.168.0.1     bridge                    0
```

Рисунок 3.6 – Маршрутна таблиця пристрою

При виконанні запитів через проткол SSH в терміналі самого маршрутизатора відображається процес авторизації (рисунок 3.6)

2023 19:54:34	memory	system, info, account	user admin logged in from 192.168.0.90 via ssh
2023 19:56:08	memory	system, info, account	user admin logged out from 192.168.0.90 via ssh
2023 19:57:22	memory	system, info, account	user admin logged in from 192.168.0.90 via telnet
2023 19:58:49	memory	system, info, account	user admin logged in from 192.168.0.90 via ssh

Рисунок 3.7 – Відображення процесу авторизації на пристрої

Конфігурація пристрою у текстовому вмгляді показна на рисунку 3.8

```
Flags: D - dynamic, X - disabled, R - running, S - slave
 0  R  name="ether1" default-name="ether1" type="ether" mtu=1500
       actual-mtu=1500 l2mtu=1598 max-l2mtu=4074
       mac-address=00:19:D1:B0:34:62 last-link-up-time=jan/01/2023 06:40:41
       link-downs=0

 1  RS name="ether2" default-name="ether2" type="ether" mtu=1500
       actual-mtu=1500 l2mtu=1598 max-l2mtu=4074
       mac-address=B8:69:F4:43:B5:DE last-link-down-time=jan/01/2023 18:31:52
       last-link-up-time=jun/01/2023 18:31:55 link-downs=10

 2   S name="ether3" default-name="ether3" type="ether" mtu=1500
       actual-mtu=1500 l2mtu=1598 max-l2mtu=4074
       mac-address=B8:69:F4:43:B5:DF link-downs=0

 3   S name="ether4" default-name="ether4" type="ether" mtu=1500
       actual-mtu=1500 l2mtu=1598 max-l2mtu=4074
       mac-address=B8:69:F4:43:B5:E0 link-downs=0

 4  RS name="ether5" default-name="ether5" type="ether" mtu=1500
       actual-mtu=1500 l2mtu=1598 max-l2mtu=4074
       mac-address=B8:69:F4:43:B5:E1 last-link-down-time=jan/01/2023 17:21:54
```

Рисунок 3.8 – Результат виконання команди /interface print detail

Тестування функції перетворення текстової інформації для аналізу конфігурації. Розбиваємо текст на окремі іменовані змінні.

```
Name: ether1
Type: ether
MTU: 1500

Name: ether2
Type: ether
MTU: 1500
```

```
Name: wlan1
Type: wlan
MTU: 1500
```

Теперд даний текст можна зберегти в файл JSON з допомогою наступної процедури.

```
entries = re.split(r"\n{2,}", text.strip())  # Розділяємо текст на окремі записи за двома порожніми рядками
data = []

for entry in entries:
    lines = entry.split("\n")
    entry_data = {}

    for line in lines:
        key, value = line.split(":")
        entry_data[key.strip()] = value.strip()

    data.append(entry_data)

json_data = json.dumps(data, indent=4)
print(json_data)
```

Цей код спочатку розділяє вхідний текст на окремі записи за допомогою порожнього рядка між ними. Потім проходиться по кожному запису і розбиває його на окремі рядки, вилучаючи ім'я і значення для створення словника. Отримані словники додаються до списку data. На останньому кроці json.dumps використовується для перетворення списку data в JSON формат з відступами (indentation) 4 пробілами.

В результаті ви отримаєте наступний JSON:

```
[    {  "Name": "ether1",
        "Type": "ether",
        "MTU": "1500"
    },
    {   "Name": "ether2",
        "Type": "ether",
        "MTU": "1500"
    },
… ]
```

JSON файли можна представити системному адміністратору для аналізу у вигляді UML діаграми (рисунок 3.9).

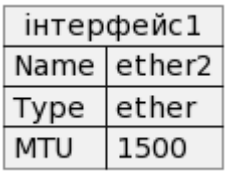

Рисунок 3.9 – Фрагмент графічного представлення конфігурації маршрутизатора

В даному розділі проведено вибір інструментальних засобів, розроблення об’єктних моделей, функцій та UML діаграм, що описують роботу програмного засобу. Здійснено реалізація та тестування програмного забезпечення.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

У даному розділі кваліфікаційної роботи проводиться економічне обґрунтування доцільності розробки програмного модуля аналізу конфігурації і правил фільтрації трафіку маршрутизатора. Зокрема, здійснюється розрахунок витрат на розробку даного програмного продукту, експлуатаційних витрат, ціни на споживання проектного рішення, визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблений електронний посібник призначений для використання студентами другого курсу спеціальності «Комп'ютерна інженерія».

4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів ($К$) включають:

$$К = К_1 + К_2, \tag{4.1}$$

де $К_1$ – витрати на розробку програмних засобів, грн.;

$К_2$ – витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($В_{ОП}$);
- витрати на відрахування у спеціальні державні фонди ($Вф$);
- витрати на покупні вироби ($Пв$);
- витрати на придбання спецобладнання для проведення експериментальних робіт ($Об$);
- накладні витрати ($Н$);

– інші витрати (*Ів*).

4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти-розробники, а саме: керівник проекту; студент-дипломник (таблиця 4.1).

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Погодинна заробітна плата, грн.
Керівник КР	124 грн/год
Студент	17 грн/год

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij}\,, \qquad (4.2)$$

де n_{ij} – чисельність розробників *i*-ої спеціальності *j*-го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником *i*-ої спеціальності *j*-го тарифного розряду, год;

C_{ij} – годинна ставка працівника *i*-ої спеціальності *j*-го тарифного розряду, грн.

Середньогодинну ставку працівника розраховуємо за формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_{i}}, \qquad (4.3)$$

де C_{ij}^{0} – основна місячна заробітна плата розробника *i*-ої спеціальності *j*-го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ – місячний фонд робочого часу працівника *i*-ої спеціальності *j*-го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку записуємо у таблицю 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год.	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
Керівник КР	16	124	1984
Студент	234	17	3978
Разом витрати на розробку			5962

4.1.2 Відрахування на соціальні заходи

Відрахування на соціальні заходи для керівника КР включають:

1) ЄСВ (єдиний соціальний внесок). Він становить 22% від заробітної плати. Оскільки заробіток керівника КР становить 1984 грн, то ЄСВ буде становити:

$$1984 \cdot 0{,}22 = 436{,}48 \text{ грн};$$

2)ПДФО (податок на доходи фізичних осіб). Він становить 18%. ПДФО буде становити:

$$1984 \cdot 0{,}18 = 357{,}12 \text{ грн};$$

3)ВЗ (військовий збір). Він становить 1,5%. ВЗ буде становити:

$$1984 \cdot 0{,}015 = 29{,}76 \text{ грн.}$$

Працівнику чистими має бути перераховано за місяць:

$$1984 - 436{,}48 - 357{,}12 - 29{,}76 = 1160{,}64 \text{ грн.}$$

Оскільки до розрахунку загального місячного (річного) оподаткованого доходу платника податку не включається, зокрема, сума стипендій України, призначених законом, постановами Верховної Ради України, указами Президента України, то заробіток студента залишається незмінним – 3978 грн.

4.1.3 Розрахунок витрат на матеріали та комплектуючі

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

Найменування	Виробник (модель)	Одиниці вимірювання	Кількість	Ціна за одиницю, грн	Сума, грн
Маршрутизатор	MikroTik RouterBOARD RB951Ui-2HnD	шт.	1	2482	2482
Разом					2482

4.1.4 Витрати на використання комп’ютерної техніки

Якщо для розробки КС використовується електрообладнання, то необхідно розрахувати витрати на електроенергію за формою, наведеною в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

Назва устаткування	Паспортна потужність	Коефіцієнт використання потужності	Час роботи обладнання, год	Ціна електроенергії, кВт год грн	Сума, грн
Ноутбук	0,17	0,8	234	1,44	57,2
Разом витрати на електроенергію					57,2

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1{,}5 \cdot 5962 = 8943 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 0{,}1 \cdot 5962 = 596{,}2 \text{ грн.}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_Ф + B_{ПВ} + H + I, \quad (4.4)$$

$$K_1 = 1847{,}6 + 378{,}76 + 187 + 2771{,}4 + 184{,}76 = 5369{,}86 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год.

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому

$$K_2 = 5{,}2 \cdot 80 = 416 \text{ грн.}$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 4.5).

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1847,6
2	Відрахування у спеціальні державні фонди	378,76
3	Витрати на куповані вироби	187
4	Накладні витрати	2771,4
5	Інші витрати	184,76
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	416
Разом		5785,52

4.2 Визначення витрат на експлуатацію програмного продукту

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Розрахуємо річні поточні витрати на експлуатацію програмного забезпечення $B_{еПК}$, які визначаються за формулою:

$$B_{еПК} = B_a + B_e + B_{рем} + B_{дк} + B_i, \quad (4.5)$$

де B_a – річні відрахування на амортизацію,

B_e – річні витрати на електроенергію для ПК,

$B_{рем}$ – річні витрати на ремонт ПК,

$B_{дк}$ – річні витрати на додаткові комплектуючі ПК,

B_i – інші витрати.

Обчислимо кожен з цих показників.

Суму річних амортизаційних відрахувань визначаємо за такою формулою:

$$B_a = Ц_{ПК} \cdot H_a, \tag{4.6}$$

де $Ц_{ПК}$ – балансова вартість ПК,

H_a – норма амортизаційних відрахувань (дорівнює 15% у квартал).

Балансову вартість ПК розраховуємо за формулою:

$$Ц_{ПК} = Ц_р \cdot (1 + K_{ун}), \tag{4.7}$$

де $Ц_р$ – ринкова вартість ПК,

$K_{ун}$ – коефіцієнт, що враховує витрати на установку й налагодження ПК (приймається рівним 12%).

Ринкова вартість ПК (Ноутбук ASUS ZenBook Duo 14 UX482EG-HY419W (90NB0S51-M003H0) Celestial Blue) становить 46000 грн. Отже, якщо $Ц_р$ = 46000, $K_{ун}$ = 0,12, то за формулою (4.7) маємо:

$$Ц_{ПК} = 46000 \cdot (1 + 0{,}12) = 51520 \text{ грн.}$$

Обчислимо норму амортизаційних відрахувань H_a. Оскільки відомо, що амортизаційні відрахування дорівнюють 15% у квартал, то

$$H_a = 4 \cdot 15\% = 60\% = 0{,}6.$$

Отже, за формулою (4.6) маємо річні відрахування на амортизацію:

$$B_a = 51520 \cdot 0{,}6 = 30912 \text{ грн.}$$

Витрати на електроенергію, що споживає ПК, визначаємо за формулою:

$$B_e = P_{ПК} \cdot Ф_{ПК} \cdot Ц_e \cdot P_{ів}, \qquad (4.8)$$

де $P_{ПК}$ – паспортна потужність ПК,

$Ф_{ПК}$ – річний фонд корисного часу роботи ПК,

$Ц_e$ – вартість 1 кВт/год електроенергії,

$P_{ів}$ – коефіцієнт інтенсивного використання ПК (0,7 – 1).

Врахувавши, що $P_{ПК}$ = 0,8, $Ф_{ПК}$ = 1843 год, $Ц_e$ = 1,44 кВт·год, $P_{ів}$ = 0,9, отримуємо:

$$B_e = 0{,}8 \cdot 1843 \cdot 1{,}44 \cdot 0{,}9 = 1911 \text{ грн.}$$

Витрати на поточний і профілактичний ремонт $B_{рем}$ приймаються рівними 6% від вартості ПК:

$$B_{рем} = Ц_{ПК} \cdot 0{,}06.$$

Отже,

$$B_{рем} = 51520 \cdot 0{,}06 = 3091{,}2 \text{ грн.}$$

Витрати на додаткові комплектуючі $B_{дк}$ – витрати необхідні для забезпечення експлуатації ПК, приймаються рівними 2% від вартості ПК:

$$B_{дк} = Ц_{ПК} \cdot 0{,}02.$$

Отже,

$$B_{дк} = 51520 \cdot 0{,}02 = 1030{,}4 \text{ грн.}$$

Інші витрати, тобто непрямі витрати пов'язані з експлуатацією ПК (приймаються рівними 5-10% від вартості ПК). Приймемо їх 5%:

$$B_i = Ц_{ПК} \cdot 0{,}05.$$

Отже,

$$B_i = 51520 \cdot 0{,}05 = 2576 \text{ грн.}$$

Отже, підставивши усі обчислені витрати у формулу (4.5), отримаємо річні поточні витрати на експлуатацію програмного забезпечення:

$$B_{еПК} = 1911 + 30912 + 3091{,}2 + 1030{,}4 + 2576 = 39520{,}6 \text{ грн.}$$

4.3 Розрахунок ціни програмного продукту

Ціна споживання програмного продукту – це витрати на придбання і експлуатацію програмного засобу за весь період його служби:

$$Ц_{С(П)} = Ц_П + B_{(E)NPV}, \quad (4.9)$$

де $Ц_П$ – ціна придбання програмного продукту, грн.

$$Ц_П = K(1 + \frac{П_Р}{100}) + K_0 + K_к,$$

де K – кошторисна вартість;

$П_р$ – рентабельність;

K_0 – витрати на прив'язку та освоєння програмного засобу на конкретному об'єкті, грн.;

$К_к$ – витрати на доукомплектування технічних засобів на об'єкті, грн. Зважаючи на вищеописане, розрахуємо ціну програмного засобу

$$Ц_П = 5785,52 \cdot (1 + 0,3) = 7521,2 \text{ грн.}$$

Вартість витрат на експлуатацію проектного продукту (за весь час його експлуатації), в грн. обчислюється так:

$$B_{enpv} = \sum_{t=0}^{T} \frac{B_{ЕП}}{(1+R)^t}, \quad (4.10)$$

де $B_{ЕП}$ – річні експлуатаційні витрати, грн.;

T – термін служби програмного засобу, років;

R – річна ставка проценту банку.

Розрахуємо витрати на експлуатацію для розробленого програмного продукту та його аналогу:

$$B_{enpv} = \sum_{t=1}^{5} \frac{4299,12}{(1+0,08)^t} = 17200,15 \text{ грн,}$$

$$B_{enpva} = \sum_{t=1}^{5} \frac{6448,68}{(1+0,08)^t} = 25800,2 \text{ грн.}$$

Тоді ціна споживання для розробленого програмного продукту та його аналогу становитиме:

$$Ц_{С(П)} = 7521,2 + 17200,15 = 24721,35 \text{ грн,}$$

$$Ц_{С(П)_a} = 6500 + 25800,2 = 32300,2 \text{ грн.}$$

У наступному підрозділі проведемо аналіз економічної ефективності розробки програмного продукту.

4.4 Визначення показників економічної ефективності

За міжнародними стандартами для оцінки ефективності розробки ПЗ застосовують такі показники:

– внутрішня норма дохідності;
– чистий приведений дохід;
– рентабельність;
– термін окупності.

Показник внутрішньої дохідності характеризує величину чистого прибутку (чистого валового доходу), що припаде на одиницю інвестиційних вкладень у кожному часовому інтервалі життєвого циклу проекту.

Розрахунок цього показника виконується за такою формулою:

$$\sum_{i=0}^{T} \frac{D_i}{(1+q)^i} - \sum_{i=0}^{T} \frac{K_i}{(1+q)^i} = 0 \qquad (4.11)$$

де D_i – дохід (прибуток) у i-му періоді;

K_i – інвестиційні вкладення в i-му періоді з урахуванням інфляційних процесів;

i – періоди виконання і впровадження проекту;

T – загальний період (тривалість) життєвого циклу проекту;

q – показник внутрішньої норми дохідності.

Показник інвестиційних вкладень з урахуванням інфляційних процесів обчислюємо за формулою:

$$K_i = \varphi_i \cdot R_i, \qquad (4.12)$$

де φ_i – коефіцієнт інфляції на поточний період;

R_i – інвестиційні платежі в i-му періоді (капітальні вкладення).

Отже,

$$K_i = 1{,}076 \cdot 70000 = 75320 \text{ грн},$$

де φ_i =107,6% (коефіцієнт інфляції поданий в таблиці на 2022 рік в сфері ІТ)

R_i = 70000 грн.

Дохід від розробки ПЗ у *i*-му періоді розраховуємо за формулою:

$$D_i = J_i\,(B_i - C_i), \qquad (4.13)$$

де B_i – ціна продажу програмного продукту в *i*-му періоді;

C_i – собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i – кількість ПЗ.

Отже,

$$D_i = 1 \cdot (80260 - 64208) = 16052 \text{ грн},$$

де B_i = 80260 грн,

C_i = 64208 грн,

J_i = 1.

Вартість продажу розробленого продукту розраховують за формулою:

$$B_i = B_{заг} \cdot (1 + p/100), \qquad (4.14)$$

де p – середній рівень рентабельності на поточний період.

Отже,

$$B_i = 64208 \cdot (1 + 25/100) = 80260 \text{ грн},$$

де p = 25%.

Показник рентабельності інвестицій. У практиці середнього бізнесу для визначення ефективності проектних рішень широко використовується показник

рентабельності інвестицій. Економічний зміст – характеризує частку чистого приведеного доходу, що припадає на одиницю дисконтованих в період життєвого циклу проекту інвестиційних вкладень:

$$p = \frac{\sum_{i=0}^{T} \frac{D_i}{(1+q)^i}}{\sum_{i=0}^{T} \frac{K_i}{(1+q)^i}} - 1 > 0. \qquad (4.15)$$

У ринкових умовах при ціновій політиці, що змінюється, показник терміну окупності є одним з головних для підприємств. Він визначається на основі величини капітальних витрат по періодах розробки програмного продукту та величини фактичних чи прогнозних доходів:

$$\sum_{i=0}^{T} K_i = \sum_{i=0}^{T} D_i, \qquad (4.16)$$

де T – термін окупності,

D_i – дохід (прибуток) у поточному періоді,

K_i – капітальні витрати у поточному періоді.

Економічна ефективність полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів:

$$E = D_i / B_{заг}.$$

Отже,

$$E = 16052 / 64208 = 0{,}15.$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = 1 / E.$$

Отже,

$$T = 1/0,15 = 6,6 \text{ років.}$$

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Враховуючи основні економічні показники, що стосуються розробки програмного продукту, можна зробити висновок, щодо доцільності запропонованої розробки. Отримано економічний ефект від розробки програмного продукту 0,15, а термін окупності капітальних вкладень 6,6 років, що є меншим 10 років, то розробка є економічно вигідною та конкурентоздатною на ринку подібних ІТ продуктів.

ВИСНОВКИ

1.Досліджено маршрутизатори, міжмережеві екрани і їх функції. Маршрутизатори, та міжмережеві екрани використовують списки контролю доступу для фільтрації пакетів на основі ідентифікаторів протоколів, IP-адрес, номерів портів та напрямку передачі. Маршрутизатори перевіряють заголовки пакетів та вирішують переслати або відкинути їх.

2. Розроблено об'єктну модель для представлення правил фільтрації і специфічні класи для різних пристроїв "TP LINK і Mikrotik. Кожен клас має свої атрибути і методи, що відповідають програмній моделі відповідного пристрою. Схема правил маршрутизаторів варіюється залежно від класу маршрутизатора, але має деякі спільні компоненти. Основними компонентами структури правила для фільтрації трафіку є умова, дія, пріоритет, напрямок, джерело та призначення.

3. Розроблено алгоритми аналізу конфігурації на основі команд ОС RouterOS для налаштування інтерфейсів, маршрутизації, файрволу, безпеки розроблено функції. Для дистанційного керування пристроєм і витягнення конфігурації використано захищений протокол SSH.

4. Програмно реалізовано розроблені алгоритми га мові Python. Розроблено узагальнену структура програмного засобу. Основні блоки: консольний інтерфейс, зчитування конфігурації з пристрою, запис конфігурації у пристрій, SSH інтерфейс маршрутизатора, редагування правил фільтрації, список правил фільтрації, перевірка коректності, візуалізація правил. Розроблено узагальнений алгоритм роботи програмного засобу.

5. Тестування програмного засобу показало коректність витягнення конфігурації маршрутизатора. Створено тестове середовище на основі маршрутизатора Mikrotik.

7. Обґрунтовано техніко-економічні показники ефективності розробки програмного забезпечення для аналізу конфігурації маршрутизатора.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Козловський О. В., Пліш В.В Засоби забезпечення безпеки мереж на основі правил. VII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 23 травня 2023 р. Тернопіль. Україна. с. 17
2. Meticlous Study of Firwall Using Secrity Detection Tools. International Jornal of Computer Aplications & Technology. 2013. Vol.2(1). p. 1-9.
3. McCabe J. Network Analysis, Architecture, and Design. Third edition. / James D. McCabe - Morgan Kaufmann, 2007. 495 p.
4. Mikrotik documentation URL: https://wiki.mikrotik.com.
5. Ileri C.U., Yigit Y., Arapoglu,O., Evcimen H.T., Asci M. Capacitated Graph Theoretical Algorithms for Wireless Sensor Networks Towards Internet of Things. Advances in Wireless Technologies and Telecommunication. 2019.
6. Venetis I.E., Gavalas, D. Pantziou, G., Konstantopoulos C. Mobile agents-based data aggregation in WSNs: benchmarking itinerary planning approaches. *Wireless Networks*, 24, 2018. p. 2111-2132.
7. "Synthesis of an Expert System for Assessing the Security of Computer Networks Based on a Fuzzy Neural Network." International Journal of Innovative Technology and Exploring Engineering. 2020.
8. Trushakov, Dmitro et al. "Basic Technical Principleыs Construction of Local Computer Systems for Manaing of Technological Objects." 2019 IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE). 2019. p.1-4.
9. The Cisco Learning Network URL: https://learningnetwork.cisco.com
10. EIA/TIA Standard, Commercial building telecommunications wiring standard (EIA/TIA-568), Electronic Industries Association, Washington, D. C., 1991.
11. Saad, A., Khan, S.A., & Mahmood, A. (2018). A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design. Swarm Evol. Comput., 38, p.187-201.
12. Zaychenko Y.P., Zaychenko H., Hamidov G. Structure optimization of new generation networks. 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017, p. 1-5.

13.Ahmed, S., & Ahmed, J. A Parallel Approach to Solve Minimum Spanning Tree Problem in Network Routing. 2017

14.Witt, R.M., & Emrich, J. Design considerations of a cable wiring system for a new medical center to support a future medical imaging system. 'Medical Imaging VI: PACS Design and Evaluation', SPIE, 1992. p. 486 -491

15.Witt, R.M., Gibbs, T., & Holden, R.W. Intercampus network of the Department of Radiology, School of Medicine, Indiana University. Medical Imaging. SPIE,1994. Vol. 2165 p. 241- 247

16.Saito, H. Theoretical Design of Geographical Route of Communications Cable Network Supplied by Power Grid to Minimize Disaster Damage. IEEE Transactions on Network and Service Management, (2022). vol.19, p.100-111.

17.Dimitri B., Robert G., "Data Networks – 2nd ed". Prentice Hall, New Jersey,

18.Gouveia L., Paixão J.P. Dynamic programming based heuristics for the topological design of local access networks. Annals of Operations Research, 1991. Vol.33, p.305-327

19. Класифікація алгоритмів маршрутизації. URL: http://um.co.ua/2/2-7/2-74318.html

20. Плешаков. CISCO Internetworking Technology Overview. URL: http://www.ods.com.ua/win/rus/net-tech/ciscoito/

21. Сімейство протколів IP. http://www.ods.com.ua/win/rus/net-tech/tcp-ip/index.htm

22. Д. Комер Міжмережевий обмін за допомогою TCP/IP. URL: http://www.ods.com.ua/win/rus/net-tech/comer/contents.htm

23. Paramiko. A Python implementation of SSHv2. URL: https://www.paramiko.org

24. https://www.technotrade.com.ua/Products/Mikrotik_RB951Ui-2HnD.php

25. Комп'ютерні мережі Частина 1 Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології», спеціалізації «Інженерія програмного забезпечення інформаційно управляючих систем» та «Інформаційне забезпечення робототехнічних систем»/ Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2020. 336 с. URL: https://ela.kpi.ua/handle/123456789/36615

26. Тарнавський Ю. А., Кузьменко І. М. Організація комп'ютерних мереж: підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення»

та 122 «Комп'ютерні науки». КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2018. 259 с. URL: https://ela.kpi.ua/handle/123456789/25156

27. Сайко В.Г., Казіміренко В.Я., Літвінов Ю.М. Мережі бездротового широкосмугового доступу. Навчальний посібник. К.: ДУТ, 2015. 196 с. URL: https://dut.edu.ua/uploads/l_881_80314158.pdf

28. Голь В.Д., Ірха М.С. Телекомунікаційні та інформаційні мережі: навчальний посібник. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського, 2021. 250 с. URL: https://ela.kpi.ua/bitstream/123456789/45409/1/TIM_navch_posib.pdf

29. The Cisco Learning Network URL: https://learningnetwork.cisco.com

30. Захист інформації в комп'ютерних системах : підручник для студ. спец. 123 «комп'ютерна інженерія» / уклад. О. М. Гапак, С. І. Балога; рец. : М. І. Глебена. – Ужгород: ПП "АУТДОР-ШАРК, 2021. – 184 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/36506

31. Eksim Ali. Wireless Communications and Networks - Recent Advances. InTech (March, 2012). 596 p. URL: https://www.intechopen.com/books/1637

32. Bonaventure Olivier. Computer Networking: Principles, Protocols and Practice. Saylor, 2022. 278 p.

33. Sandy Hirtz Education for a Digital World: Advice, Guidelines, and Effective Practice from Around the Globe. 2008. 516 p. URL: http://www.colfinder.org/materials/Education_for_a_Digital_World/Education_for_a_Digital_World_complete.pdf

34. Палеха Ю. І. Етика ділових відносин: Навч. посіб. К.: Кондор, 2007. 356с. URL: https://library.nlu.edu.ua/POLN_TEXT/KNIGI/KONDOR1/CD/ETUKA_DV.pdf

35. Шкіцька І. Ю. Основи академічної доброчесності: практикум: навчально-методичний посібник для студентів вищих навчальних закладів. Тернопіль: ТНЕУ, 2018. 64 с.

36. Computer Engineering Curricula 2016 URL: https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf

37. ДСТУ 3973-2000 «Правила виконання науково-дослідних робіт. Загальні положення»

38. Camisso Jamon. Making Servers Work: A Practical Guide to Linux System Administration. DigitalOcean, 2020. 281 p. URL: https://www.digitalocean.com/community/books/sysadmin-ebook-making-servers-work

39. ДСТУ 3719:1998 (ISO/IEC 8613:1989) Інформаційні технології. Електронний документообіг. Архітектура службових документів (ODA) та обмінний формат. Частини 1-4

40. Kline, Kappos. Introduction to Intellectual Property. 2021 URL: https://open.umn.edu/opentextbooks/textbooks/introduction-to-intellectual-property

41. Shotts W. The Linux Command Line: A Complete Introduction. 5 ed. 2019. 555 p. URL: https://linuxcommand.org/tlcl.php

42. Neil Smyth Ubuntu 20.04 Essentials: A Guide to Ubuntu Desktop and Server. 2020. URL: https://www.answertopia.com/ubuntu/ubuntu-essentials/

43. Мулеса О.Ю. Основи мови запитів SQL. Ужгород, 2015. 48 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/8868/1/sql.pdf

44. Apache HTTP Server Version 2.5 Documentation URL: https://httpd.apache.org/docs/trunk/

45. Carlos De La Guardia. Python Web Frameworks. O'Reilly Media, Inc. 2016. URL: https://www.oreilly.com/content/python-web-frameworks/

46. ДСТУ 3008:2015 Національний стандарт України. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Введ. 01.07.2017. К.: ДП "УкрНДНЦ, 2016. 25 с

47. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Введ. 01.07.2016. К.: ДП «УкрНДНЦ», 2017. 16 с.

48. Методичні вказівки до випускних кваліфікаційних робіт освітнього рівня “Бакалавр” спеціальності “Комп’ютерна інженерія”/ О.М. Березький, Г.М. Мельник, Л.О.Дубчак, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2021. – 52 с.

49. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп’ютерних систем”/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

50. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.