МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри Комп'ютеризованих
систем захисту інформації

_____ Михайло СТЕПАНОВ
«____»_____ 2023 р.

На правах рукопису
УДК 004.056.5:510.22(043.3)

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль керування доступом до інформаційних ресурсів

Виконавець: Володимир ПАРХОМЕНКО

Керівник: д.т.н., професор Сергій ТОЛЮПА

Консультант розділу "Охорона навколишнього середовища" Тетяна ДМИТРУХА

Нормоконтролер: д.т.н., професор Сергій ТОЛЮПА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ

«___» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

здобувача вищої освіти Пархоменка Володимира

1) Тема роботи " Програмний модуль керування доступом до інформаційних ресурсів/ Software module for managing access to information resources" затверджена наказом ректора від «__» _____ 20__ № _____/ст..
2) Термін виконання з __.__.20__р. по __.__.20__р.
3) Перелік питань, що належить дослідити в процесі написання роботи:
   a) Дослідити доступ до інформаційних ресурсів через локальну мережу за допомогою технології VPN.
   b) Дослідити доступ до інформаційних ресурсів технологією RDP (Remote desktop protocol).
4) Завдання на роботу написати: "Програмний модуль керування доступом до інформаційних ресурсів "

5. КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ з/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	Виконано
2.	Аналіз літературних джерел	18.10.2023	Виконано
3.	Обґрунтування вибору рішення	20.10.2023	Виконано
4.	Збір інформації	22.10.2023	Виконано
5.	Нормативно-правова складова кібербезпеки	24.10.2023	Виконано
6.	Загальна характеристика управління доступом	02.11.2023	Виконано
7.	Аналіз доступу до інформаціних ресурсів	10.11.2023	Виконано
8.	Оцінка методів доступу до інформаційних ресурсів	11.11.2023	Виконано
9.	Практична реалізація програмного модуля керування доступом до інформаційних ресурсів	20.11.2023	Виконано
10.	Апробація роботи	02.12.2023	Виконано
11.	Перевірка на антиплагіат	08.12.2023	Виконано
12.	Оформлення і друк пояснювальної записки	10.12.2023	Виконано
13.	Оформлення презентації	14.12.2023	Виконано
14.	Отримання рецензій	22.12.2023	Виконано

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

7. Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Володимир ПАРХОМЕНКО

(підпис)

Керівник кваліфікаційної роботи Сергій ТОЛЮПА

(підпис)

РЕФЕРАТ

Магістерська кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків і має 45 сторінок основного тексту, 17 рисунків, 1 таблицю, 1 додаток. Список використаних джерел містить 20 найменування. Загальний обсяг роботи сторінок 75.

Мета і завдання дослідження. створення програмного модуля керування доступом до інформаційних ресурсів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– Дослідити доступ до інформаційних ресурсів через локальну мережу за допомогою технології VPN;

– Дослідити доступ до інформаційних ресурсів технологією RDP (Remote desktop protocol)

– Реалізувати програмний модуль керування доступом до інформаційних ресурсів.

Об’єкт дослідження – є процес роботи програмного модуля взаємодії з інформаційними ресурсами.

Предмет дослідження – є методи, моделі та системи керування віддаленим доступом інформаційних ресурсів.

Новизна - написано програмний модуль управління доступом до інформаційних ресурсів, що дозволяє спрощувати роботу IT відділів та системних адміністраторів, , а також забезпечити безперервний доступ працівника до інформаційних ресурсів компанії

Практична цінність полягає у розробці програмного модуля управління доступом до інформаційних ресурсів. Завдяки данному додатку можна значно

спростити роботу компаній та дата центрів, а також забезпечити безперервний доступ працівника до інформаційних ресурсів компанії.

Ключові слова: айпі, програмний модуль керування доступом, сервер, користувач, системний адміністратор, айті відділ, інформаційні ресурси.

Апробація

Сергій Толюпа, Володимир Пархоменко. Керування доступом до інформаційних ресурсів. Збірник матеріалів доповідей та тез VI міжнародної науково-практичнної конференції «Проблеми кібербезпеки інформаційно-телекомунікаційних систем» м. Київ, 27 квітня 2023 року. – К.: ВПЦ "Київський університет", 2023. – С. 64-66.

ЗМІСТ

Перелік умовних скорочень 7

Вступ 8

Розділ 1. Управління доступом 10

1.1. Принципи безпеки 11
1.1.1. Доступність 12
1.1.2. Цілісність 13
1.1.3. Конфіденційність 14
1.2. Ідентифікація 15
1.3. Логічне управління доступом 16
1.4. Аутентифікація 17
1.5. Авторизація 18
1.6. Аудит і моніторинг 19
1.7. Контроль доступу 21

Розділ 2. Теоретичні відомості доступу до інформаційних ресурсів 22

2.1.. Доступ до інформаційних ресурсів засобами VPN 22
2.2. Класифікація VPN 24
2.3. Доступ до інформаційних ресурсів засобами RDP 28
2.4. NAT - Network Address Translation 30
2.5. Типи NAT 34
2.6. Microtik 35
2.7. NAS 37
2.8. RDS 38
2.9. MSSQL 39
2.10. CRM 41

Висновок до другого розділу 43

Розділ 3. Розробка програмного модуля керування доступом до інформаційний ресурсів. 44

3.1. Актуальність розробки 44

3.2. Розробка та робота програмного модуля керування доступом до інформаційних ресурсів 44

Розділ 4. Охорона навколишнього середовища 53

4.1. Заповідні території в Україні 53

Висновки 62
Список використаної літератури 63
Додаток А 65

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

RDP	• Remote Desktop Protocol;
VPN	• Virtual Private Network;
NAT	• Network Address Translation;
PPtp	• Point-to-Point Tunneling Protocol;
ПК	• персональний компьютер;
L2tp	• Layer 2 Tunnelling Protocol;
IP	• Internet Protocol;
IT	• технічний відділ;
TCP	• Transmission Control Protocol;
UDP	• User Datagram Protocol;
Wi-Fi	• технологія бездротової локальної мережі
RDS	• Remote Desktop Services
CRM	• Customer Relationship Management
MSSQL	• Microsoft SQL Server

ВСТУП

Стрімкий розвиток інформаційних систем, комп'ютеризація та впровадження новітніх технологій в усі сфери діяльності суспільства і держави в цілому, значно прискорює і удосконалює роботу організацій, підприємств та установ усіх форм власності. Однак, такий розвиток несе в собі цілий ряд загроз пов'язаних з порушеннями конфіденційності, цілісності та доступності нформації, які, в свою чергу, призводять до різних втрат (в тому числі - фінансовим) і часто - досить значним.

За останні роки все частіше використовуються інформаційні ресурси і системи, як інструмент задоволення потреб. У даному випадку маються на увазі не тільи інформаційні потреби, а й потреби у багатьох головних сферах людської діяльності – бізнес потреби, економічні потреби, громадські, індивидуальні та інші. Активне використання інформаційних ресурсів в мережі Інтернет, у свою чергу створює додаткові загрози.

Рішення ж питань безпеки буде успішним тільки за умови використання комплексного підходу до побудови систем захисту.

В даній роботі ми розглянемо способи доступу до інформаційних ресурсів та побудуємо зручний механізм вирішення досить поширеної проблеми відсутності часу системних адміністраторів на обробку тих чи інших запитів користувачів, в тому числі в будь який проміжок часу, таким чином полегшимо роботу створивши додаток який дозволить допомогти у вирішенні даної проблеми.

Актуальність. На даний момент досить важливо отримувати своєчасний доступ до інформаційних ресурсів, в такому випадку працівник може зробити свою роботу як віддалено так і в офісі. Саме для першого варіанту і підходить наша

розробка, проте вона вузько спеціалізована і протестована лише з обладнання фірми microtik.

Метою роботи є створення програмного модуля керування доступом до інформаційних ресурсів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– Дослідити доступ до інформаційних ресурсів через локальну мережу за допомогою технології VPN;

– Дослідити доступ до інформаційних ресурсів технологією RDP (Remote desktop protocol);

– Реалізувати програмний модуль керування доступом до інформаційних ресурсів.

Галузь застосування. Дана розробка може бути застосована в роботі невеликих та середніх компаній, а також невеликих та середніх дата центрів, що використовують в своїй роботі обладнання фірми microtik.

Об'єктом дослідження є процес роботи програмного модуля взаємодії з інформаційними ресурсами.

Предметом дослідження є методи, моделі та системи керування віддаленим доступом інформаційних ресурсів.

Новизна одержаних результатів полягає в наступному:

написано програмний модуль управління доступом до інформаційних ресурсів, що дозволяє спрощувати роботу IT відділів та системних адміністраторів.

Розділ 1. УПРАВЛІННЯ ДОСТУПОМ

Управління доступом - це механізм безпеки, який керує процесом взаємодії користувачів з системами і ресурсами, а також систем між собою. Цей механізм захищає системи і ресурси від несанкціонованого доступу і бере участь в визначенні рівня авторизації після успішного проходження процедури аутентифікації. Не можна забувати про те, що крім користувачів, в мережі існують і інші суб’єкти, яким потрібен доступ до мережевих ресурсів і інформації. У процесі управління доступом необхідно знати і розуміти визначення суб'єкта та об'єкта.

Контроль доступу відноситься до вибіркового обмеження доступу до ресурсу або системи. Це техніка безпеки, яка використовується для регулювання того, хто або що може переглядати або використовувати ресурси в обчислювальному середовищі. Контроль доступу має вирішальне значення для підтримки конфіденційності, цілісності та доступності даних і ресурсів в інфраструктурі організації.[20]

Існують різні види контролю доступу:

1) Дискреційний контроль доступу (DAC): Ця модель дозволяє власнику ресурсу контролювати доступ до цього ресурсу та визначати, хто може отримати до нього доступ і які дозволи вони мають. Це на власний розсуд, оскільки власник має право доступу щодо прав доступу;

2) Обов’язковий контроль доступу (MAC): у цій моделі засоби контролю доступу визначаються центральним органом або системним адміністратором на основі політик безпеки. Доступ ґрунтується на дозволах безпеки та мітках, призначених як користувачам, так і ресурсам;

3) Контроль доступу на основі ролей (RBAC): доступ надається на основі ролей користувачів в організації. Дозволи пов’язані з ролями, а користувачам призначаються ролі, які визначають їхні права доступу;

4) Контроль доступу на основі атрибутів (ABAC): доступ визначається оцінкою атрибутів, пов’язаних із користувачем, ресурсом і середовищем. Ці атрибути використовуються для прийняття рішень щодо динамічного контролю доступу.

Механізми контролю доступу можуть бути реалізовані за допомогою різних технологій, таких як списки контролю доступу (ACL), методи автентифікації користувачів (паролі, біометрія тощо), шифрування, брандмауери тощо. Ці механізми використовуються на різних рівнях інфраструктури організації, включаючи контроль фізичного доступу до будівель, контроль доступу до мережі, контроль доступу до файлів в операційних системах і контроль доступу до бази даних.[18]

Ефективний контроль доступу необхідний для запобігання несанкціонованому доступу, захисту конфіденційної інформації та забезпечення того, щоб користувачі мали відповідний рівень доступу, необхідний для виконання своїх ролей без шкоди для безпеки.

1.1. Принципи безпеки

Існує три основних принципи безпеки для будь-яких видів управління безпекою: доступність, цілісність і конфіденційність. Кожен механізм захисту (або управління) реалізує як мінімум один з цих принципів. Спеціаліст з безпеки повинен розуміти всі можливі способи реалізації цих принципів.

1.1.1. Доступність

Доступність означає розробку та впровадження продуктів, послуг, середовищ або систем, якими можуть користуватися люди з різними здібностями, у тому числі з обмеженими можливостями. Мета доступності полягає в тому, щоб кожен, незалежно від своїх фізичних або когнітивних здібностей, міг отримати доступ і використовувати однакову інформацію та ресурси.[19]

У контексті технологій і цифрового вмісту доступність зосереджена на тому, щоб зробити веб-сайти, програми, програмне забезпечення та цифровий вміст придатними для використання людьми з обмеженими можливостями. Це включає, але не обмежується:

1) Візуальна доступність: розробка вмісту, до якого можуть отримати доступ сліпі люди, які мають слабкий зір або дальтонізм. Це передбачає використання належного контрасту, надання текстових альтернатив для зображень і забезпечення сумісності з програмами зчитування з екрана;

2) Доступність моторики та спритності: створення інтерфейсів доступними для людей з обмеженою мобільністю або тих, хто використовує допоміжні пристрої. Це може передбачати надання комбінацій клавіш, можливість настроюваних елементів керування та уникнення обмежених у часі взаємодій;

3) Слуховий доступ: забезпечення того, щоб вміст був доступним для людей із вадами слуху або слуху. Це включає надання субтитрів для відео, розшифровок для аудіовмісту та візуальних підказок для звуків;

4) Когнітивна доступність: розробка контенту, зрозумілого та зручного для людей із когнітивними порушеннями. Це може включати чітку мову, узгоджену навігацію та уникнення блимаючих або відволікаючих елементів.

Щоб досягти доступності, організації часто дотримуються стандартів і вказівок, таких як Рекомендації щодо доступності веб-контенту (WCAG), розроблені Консорціумом всесвітньої павутини (W3C). Ці вказівки містять набір критеріїв і рекомендацій щодо того, як зробити цифровий контент доступним для широкого кола користувачів.[20]

Створення доступних технологій не тільки приносить користь людям з обмеженими можливостями, але й покращує зручність використання для всіх, створюючи більш інклюзивний і різноманітний досвід користувачів.

1.1.2. Цілісність

Цілісність у контексті даних або інформаційних систем означає якість точності, узгодженості та надійності даних протягом усього життєвого циклу. Це гарантує, що дані залишаються недоторканими та незмінними порівняно з їх початковим станом, зберігаючи їхню надійність і достовірність. Є кілька аспектів цілісності даних:

1) Точність: дані вважаються точними, якщо вони відображають реальні цінності, які вони повинні представляти. Точність гарантує, що інформація є правильною та не містить помилок або розбіжностей;

2) Узгодженість: узгодженість даних означає одноманітність і стабільність інформації. Послідовні дані залишаються незмінними в різних базах даних, програмах або в певні моменти часу;

3) Надійність: Надійним даним можна довіряти та покладатися на них для прийняття рішень і операцій. Він є послідовним, точним і залишається незмінним, якщо його навмисно не змінено за допомогою авторизованих процесів;

4) Дійсність: дійсні дані відповідають визначеним правилам і обмеженням. Він відповідає певним форматам, діапазонам або умовам, щоб вважати його прийнятним.

Підтримка цілісності даних передбачає впровадження заходів для запобігання несанкціонованим або випадковим змінам, забезпечення захисту даних від пошкодження або несанкціонованого доступу. Це включає використання засобів контролю доступу, шифрування, резервного копіювання та процесів перевірки для захисту даних від порушень цілісності.[19]

Цілісність даних є фундаментальною для забезпечення якості та достовірності інформації в базах даних, системах і програмах. Це критично важливо в різних секторах, включаючи фінанси, охорону здоров’я та будь-яку іншу галузь, де точна та надійна інформація є важливою для прийняття рішень і операцій.

1.1.3. Конфіденційність

Конфіденційність означає право людини контролювати доступ до своєї особистої інформації та визначати, як ця інформація збирається, використовується, ділиться та зберігається. Це передбачає захист конфіденційних даних від несанкціонованого доступу, розголошення або неправомірного використання.[18]

У сучасну цифрову епоху конфіденційність стала серйозною проблемою через величезну кількість особистих даних, які збираються, зберігаються та обробляються різними організаціями, зокрема підприємствами, урядами та онлайн-платформами. Основні аспекти конфіденційності включають:

1) Збір даних: це передбачає збір особистої інформації за допомогою різних засобів, таких як форми, дії в Інтернеті, транзакції та взаємодії. Прихильники конфіденційності наголошують на необхідності прозорої та законної практики збору, отримання згоди та обмеження збору даних необхідним;

2) Обмеження використання даних і цілей: організації повинні використовувати зібрані дані лише для визначених і законних цілей. Особиста інформація не повинна використовуватися чи оброблятися у спосіб, який не відповідає початковій меті, без отримання додаткової згоди;

3) Безпека даних. Захист особистої інформації від несанкціонованого доступу, злому чи крадіжки має вирішальне значення. Застосування заходів безпеки, таких як шифрування, контроль доступу та регулярні перевірки безпеки, допомагає захистити конфіденційні дані;

4) Обмін даними та їх розкриття. Організації мають надавати особисту інформацію третім особам лише за потреби, і вони повинні робити це відповідально та безпечно. Користувачі повинні бути проінформовані про такий обмін і мати можливість контролювати його, коли це можливо.

5) Контроль користувача та прозорість: особи повинні мати право доступу до власних даних, виправляти неточності та контролювати, як використовується їхня

інформація. Прозорість щодо практики обробки даних, політики та причин збору даних має важливе значення для зміцнення довіри.

Такі нормативні акти, як Загальний регламент захисту даних (GDPR) у Європі та Каліфорнійський закон про конфіденційність споживачів (CCPA) у Сполучених Штатах, є прикладами законодавчих заходів, спрямованих на захист прав особи на конфіденційність і накладення зобов'язань на організації щодо обробки даних і згоди користувачів.

Захист конфіденційності не лише поважає права людей, але й сприяє довірі між користувачами та організаціями, що веде до більш етичних і стійких відносин у цифровій екосистемі.

1.2. Ідентифікація

Ідентифікація — це процес встановлення або підтвердження особистості особи. Це передбачає надання облікових даних, інформації або характеристик, які однозначно відрізняють одну особу від іншої. Для ідентифікації використовуються різні методи та фактори:

1) Облікові дані або документи: це включає використання офіційних посвідчень особи, паспортів, водійських прав або інших офіційних документів, які містять особисту інформацію та фотографію для підтвердження особи;

2) Біометрична ідентифікація: біометрія використовує унікальні фізичні характеристики або поведінкові риси, такі як відбитки пальців, сканування райдужної оболонки ока, розпізнавання обличчя, голосові шаблони або навіть ДНК, щоб підтвердити особу;

3) Ідентифікація на основі знань: передбачає використання інформації, яку знала б лише ідентифікована особа, як-от паролі, PIN-коди, секретні запитання або особисті дані, як-от дати народження чи адреси;

4) Жетони або пристрої. Ідентифікація також може включати використання фізичних маркерів, таких як смарт-карти, брелоки або електронні пристрої, які зберігають або генерують унікальні коди чи облікові дані для підтвердження особи.

Ідентифікація є фундаментальною в різних контекстах, зокрема для доступу до безпечних місць, перевірки права на послуги чи переваги, запобігання шахрайству, захисту облікових записів в Інтернеті та забезпечення конфіденційності та безпеки в різних транзакціях.

Ефективні та безпечні методи ідентифікації мають вирішальне значення для захисту конфіденційності людей, запобігання крадіжці особистих даних і забезпечення цілісності систем і служб, які покладаються на підтвердження та перевірку ідентичності користувачів.[20]

1.3. Логічне управління доступом

Логічний контроль доступу відноситься до заходів безпеки та політик, реалізованих для управління та контролю доступу до комп'ютерних систем, мереж і цифрових ресурсів. Він зосереджений на регулюванні прав електронного доступу користувачів до інформаційних систем організації.

Ключові компоненти логічного контролю доступу включають:

1) Автентифікація: процес перевірки ідентичності користувача або організації, які намагаються отримати доступ до системи або ресурсу. Це може включати такі облікові дані, як імена користувачів, паролі, біометричні дані, смарт-карти або токени;

2) Авторизація: після автентифікації користувача авторизація визначає конкретні дії та ресурси, до яких користувачеві дозволено доступ або використання. Це передбачає призначення дозволів, ролей або рівнів доступу на основі ролі або привілеїв користувача;

3) Контроль доступу: механізми, які забезпечують виконання політик і обмежень контролю доступу. Це може включати брандмауери, системи виявлення вторгнень, шифрування, списки контролю доступу (ACL) та інші заходи безпеки;

4) Аудит і моніторинг: моніторинг і журнал дій користувачів і спроб доступу з метою аудиту. Це допомагає відстежувати та переглядати події доступу, виявляти потенційні порушення безпеки та забезпечувати відповідність політикам безпеки.

Логічний контроль доступу необхідний для захисту конфіденційної інформації, запобігання несанкціонованому доступу або витоку даних, а також забезпечення того, щоб користувачі мали доступ лише до ресурсів, необхідних для виконання їхніх ролей або обов'язків в організації. Це важливий аспект загальних стратегій кібербезпеки, особливо в середовищах, де кілька користувачів мають доступ до взаємопов'язаних систем і даних.

1.4. Аутентифікація

Автентифікація — це процес перевірки особи або системи, які намагаються отримати доступ до певного ресурсу, наприклад комп'ютерної системи, мережі, програми або даних. Це гарантує, що суб'єкт, який намагається отримати доступ, дійсно є тим, ким або ким він себе видає.

Існують різні методи автентифікації, зокрема:

1) Паролі та парольні фрази: користувачі надають секретну комбінацію символів, яку вони знають, наприклад пароль або парольну фразу;

2) Біометрична автентифікація: це передбачає використання унікальних фізичних характеристик, таких як відбитки пальців, розпізнавання обличчя, сканування райдужної оболонки ока або голосові шаблони для підтвердження особи;

3) Автентифікація на основі маркерів: використання фізичних пристроїв (токенів) або згенерованих програмним забезпеченням маркерів, які генерують тимчасові коди для доступу, як-от одноразові паролі (OTP) ;

4) Багатофакторна автентифікація (MFA): поєднання двох або більше різних методів автентифікації (наприклад, пароль + SMS-код, відбиток пальця + смарт-карта) для підвищення безпеки;

5) Автентифікація на основі сертифіката: використання цифрових сертифікатів, виданих довіреним органом, для перевірки ідентичності користувача або системи.

Автентифікація гарантує, що лише авторизовані особи або системи можуть отримати доступ до конфіденційних даних або ресурсів, що значно сприяє заходам безпеки як у фізичному, так і в цифровому середовищах. Впровадження надійних методів автентифікації має вирішальне значення для захисту від несанкціонованого доступу, витоку даних та інших загроз безпеці.[19]

1.5. Авторизація

Авторизація — це процес надання або заборони доступу до ресурсів або функціональних можливостей на основі ідентичності користувача, дозволів або атрибутів після успішної автентифікації. Коли особу користувача підтверджено, авторизація визначає, які дії йому дозволено виконувати та до яких даних або ресурсів він може отримати доступ у системі чи програмі.

Ключові компоненти авторизації включають:

1) Дозволи та контроль доступу: призначення конкретних дозволів або привілеїв користувачам або групам користувачів на основі їхніх ролей, обов’язків або рівнів повноважень в організації. Це передбачає визначення того, які дії (читання, запис, видалення, виконання тощо) користувачі можуть виконувати на певних ресурсах;

2) Політики доступу: встановлення правил і політик, які визначають, хто може отримати доступ до яких ресурсів і за яких умов. Ці політики часто визначаються адміністраторами та забезпечуються механізмами контролю доступу;

3) Контроль доступу на основі ролей (RBAC): Організація користувачів у ролі та призначення дозволів на основі цих ролей. Наприклад, працівник на керівній посаді може мати доступ до певних конфіденційних даних або функцій, яких немає у звичайного працівника;

4) Контроль доступу на основі атрибутів (ABAC): прийняття рішень щодо доступу на основі різних атрибутів або характеристик, пов'язаних з користувачем, наприклад його відділу, місця розташування, посади або конкретних атрибутів користувача.

Механізми авторизації працюють рука об руку з автентифікацією, щоб гарантувати, що автентифіковані користувачі мають доступ лише до ресурсів або функцій, які їм дозволено використовувати. Правильне впровадження елементів керування авторизацією має вирішальне значення для підтримки безпеки даних, запобігання несанкціонованому доступу та забезпечення дотримання правил конфіденційності та політики організації.

1.6. Аудит і моніторинг

Аудит і моніторинг є важливими компонентами комплексної стратегії кібербезпеки, зокрема щодо підтримки безпеки, цілісності та відповідності систем, мереж і даних. Ці процеси включають відстеження, спостереження та перегляд дій, подій або доступу в IT-середовищі, щоб забезпечити ефективність і дотримання заходів безпеки.

Ось як аудит і моніторинг відіграють вирішальну роль у кібербезпеці:

1. Журнал подій: системи створюють журнали, які реєструють різні події, такі як спроби входу, доступ до файлів, системні зміни та мережева діяльність. Ці журнали служать для запису того, що сталося в системі, і є ключовими для аудиту та аналізу;

2. Моніторинг у режимі реального часу: постійний моніторинг систем і мереж у режимі реального часу допомагає виявляти підозрілі дії, аномалії чи

потенційні порушення безпеки, коли вони відбуваються. Для цього використовуються системи виявлення вторгнень, інструменти моніторингу мережі та системи управління інформацією та подіями безпеки (SIEM);

3. Відповідність і регулювання. Аудит і моніторинг допомагають забезпечити відповідність систем і методів галузевим нормам, стандартам відповідності та політикам внутрішньої безпеки. Для перевірки дотримання цих стандартів проводяться регулярні аудити;

4. Реагування на інциденти та криміналістика: у разі інциденту безпеки журнали аудиту та дані моніторингу мають вирішальне значення для розслідування причини, розуміння масштабу порушення та проведення судово-медичного аналізу для запобігання майбутнім інцидентам;

5. Відстеження активності користувачів: моніторинг активності користувачів допомагає виявити несанкціоновані дії або підозрілу поведінку. Це включає відстеження часу входу, доступу до файлів, зміни дозволів та інших дій, пов’язаних із користувачем;

6. Попередження та звітування: системи можуть бути налаштовані на створення попереджень або сповіщень у разі порушення певних попередньо визначених подій або порогів. Звіти, створені на основі даних аудиту та моніторингу, дають уявлення про стан системи, потенційні вразливості та тенденції безпеки.

Регулярно переглядаючи та аналізуючи журнали аудиту та дані моніторингу, організації можуть завчасно виявляти слабкі місця безпеки, оперативно реагувати на інциденти безпеки та покращувати загальний стан кібербезпеки. Ця практика допомагає підтримувати конфіденційність, цілісність і доступність критично важливих систем і даних.[18]

1.7. Контроль доступу

Контроль доступу може бути реалізований на різних рівнях:

1) Контроль фізичного доступу: регулювання входу до фізичних приміщень, таких як будівлі, кімнати або центри обробки даних, за допомогою таких механізмів, як ключ-картки, біометричні сканери або охоронці;

2) Логічний контроль доступу: Управління доступом до цифрових ресурсів, таких як комп’ютерні системи, мережі, бази даних і програми, за допомогою автентифікації, авторизації та інших заходів безпеки;

Ефективний контроль доступу має вирішальне значення для захисту конфіденційних даних, запобігання несанкціонованому доступу або порушенням, забезпечення відповідності нормам і підтримки загальної безпеки активів та інформації організації.

Розділ 2. Теоретичні відомості доступу до інформаційних ресурсів

2.1 Доступ до інформаційних ресурсів засобами VPN

Отже почнемо, в сучасному світі велику роль в допомозі розвитку підприємства відіграють саме інформаційні ресурси. Цими ресурсами можуть бути сайти, бази даних, сервери та інші пристрої, створені для полегшення виконання завдань прибутку та оптимізації процесів в компаніях. Проте ці ресурси потребують захисту, в еру розвитку штучного інтелекту, нейромереж та стрімкого скачка IT індустрії ми не можемо бути впевнені в захисті персональних даних, а також даних так званих робочих, саме тих документів що ми використовуємо на роботі, таблиць списків та інших, і нам треба думати про захист.

В даній роботі я опишу дві технології, які використовують для захисту ресурсів на даний момент, звичайно в кожній є свої недоліки а також свої нюанси налаштування, проте таким чином ми можемо більш детально вивчити кожну та запропонувати нашому роботодавцю найбільш необхідну та підходящу до його типу підприємства.

Переходимо до першого варіанту: Даний варіант пропонує об’єднувати всі комп’ютери підприємства в одну велику мережу, навіть якщо працівник знаходиться не в офісі, за допомогою VPN він зможе доєднатися до необхідної локальної мережі та відповідно мати доступ до тих ресурсів які необхідні.

VPN або віртуальні приватні мережі — це інструменти, які створюють безпечне з’єднання між вашим пристроєм та Інтернетом.[8] Вони шифрують ваш інтернет-трафік, роблячи його більш безпечним і приватним. Люди часто використовують VPN з різних причин:

1) Безпека: мережі VPN шифрують ваші дані, запобігаючи їх перехопленню хакерами або сторонніми особами, особливо під час використання публічних мереж Wi-Fi;

2) Конфіденційність: вони приховують вашу IP-адресу та місцезнаходження, що ускладнює відстеження ваших дій в Інтернеті для веб-сайтів, рекламодавців або навіть вашого провайдера (провайдера Інтернет-послуг) ;

3) Доступ: мережі VPN можуть надавати доступ до геообмеженого вмісту, роблячи так, ніби ви отримуєте доступ до Інтернету з іншого місця. Наприклад, ви можете отримати доступ до вмісту, який може бути обмежений у вашій країні;

4) Обхід цензури: у деяких країнах певні веб-сайти чи служби можуть бути заблоковані. VPN можуть допомогти обійти ці обмеження, маршрутизуючи ваше з'єднання через сервери в інших країнах із більш відкритою політикою Інтернету.[10]

Однак, незважаючи на те, що VPN пропонують підвищену конфіденційність і безпеку, важливо вибрати надійного постачальника послуг. Крім того, майте на увазі, що VPN можуть дещо сповільнити ваше інтернет-з'єднання через шифрування та додаткову маршрутизацію сервера. [1]

Користуючись VPN, переконайтеся, що ви ознайомилися з політикою конфіденційності постачальника послуг, оскільки деякі можуть реєструвати дані користувача, незважаючи на те, що вони цього не повинні робити. [1]

Існують різні типи впн які ми можемо використати для забезпечення роботи працівників, це впн між маршрутизаторами або ми також можемо вибрати тип клієнт – маршрутизатор. В обох випадках створюється тунель завдяки якому користувач може підключатися з будь якої мережі в корпоративну та відповідно до правил NAT мати доступ до тих чи інших ресурсів компанії.

Також ми можемо вибрати різні протоколи (PPTP, L2tp, OpenVPN) все залежить від того для чого ми вибираємо той чи інший впн - для якого підключення.

Наприклад протокол PPTP вважається вже застарілим і відповідно його не бажано використовувати. Різні ресурси що надають можливість використовувати VPN як сервіс поступово відмовляються від PPTP серверів на користь L2tp.

Більш конкретно VPN складається з двох частин: «внутрішня» (підконтрольна) мережа, яких може бути кілька, і «зовнішня» мережа, через яку проходять інкапсульовані з'єднання (зазвичай використовується Інтернет).

Можливо також під'єднання до віртуальної мережі окремого комп'ютера.[1]

2.2.Класифікація VPN

VPN класифікують за типом використовуваного середовища таким чином:

Захищені: Найпоширеніший варіант віртуальних приватних мереж. З його допомогою можливо створити надійну і захищену підмережу на основі ненадійної мережі, зазвичай, Інтернету. Прикладом захищених протоколів VPN є: Ipsec, SSL та PPTP. Прикладом використання протоколу SSL є програмне забезпечення OpenVPN.

Довірчі: Використовують у випадках, коли середовище, яким передають дані, можна вважати надійним і потрібно розв'язати лише завдання створення віртуальної підмережі в рамках більшої мережі. Питання забезпечення безпеки стають неактуальними. Прикладами подібних VPN рішень є: Multi-protocol label switching (MPLS) і L2tp (Layer 2 Tunnelling Protocol). (Коректніше сказати, що ці протоколи перекладають завдання забезпечення безпеки на інших, наприклад L2tp, як правило, використовують разом з Ipsec).[3]

Зазвичай VPN утворюють на рівнях не вище мережевого, бо застосування криптографії на цих рівнях дозволяє використовувати в незмінному вигляді транспортні протоколи (такі як TCP, UDP). Користувачі Microsoft Windows позначають терміном VPN одну з реалізацій віртуальної мережі — PPTP, причому вона частіше використовується не для створення приватних мереж.

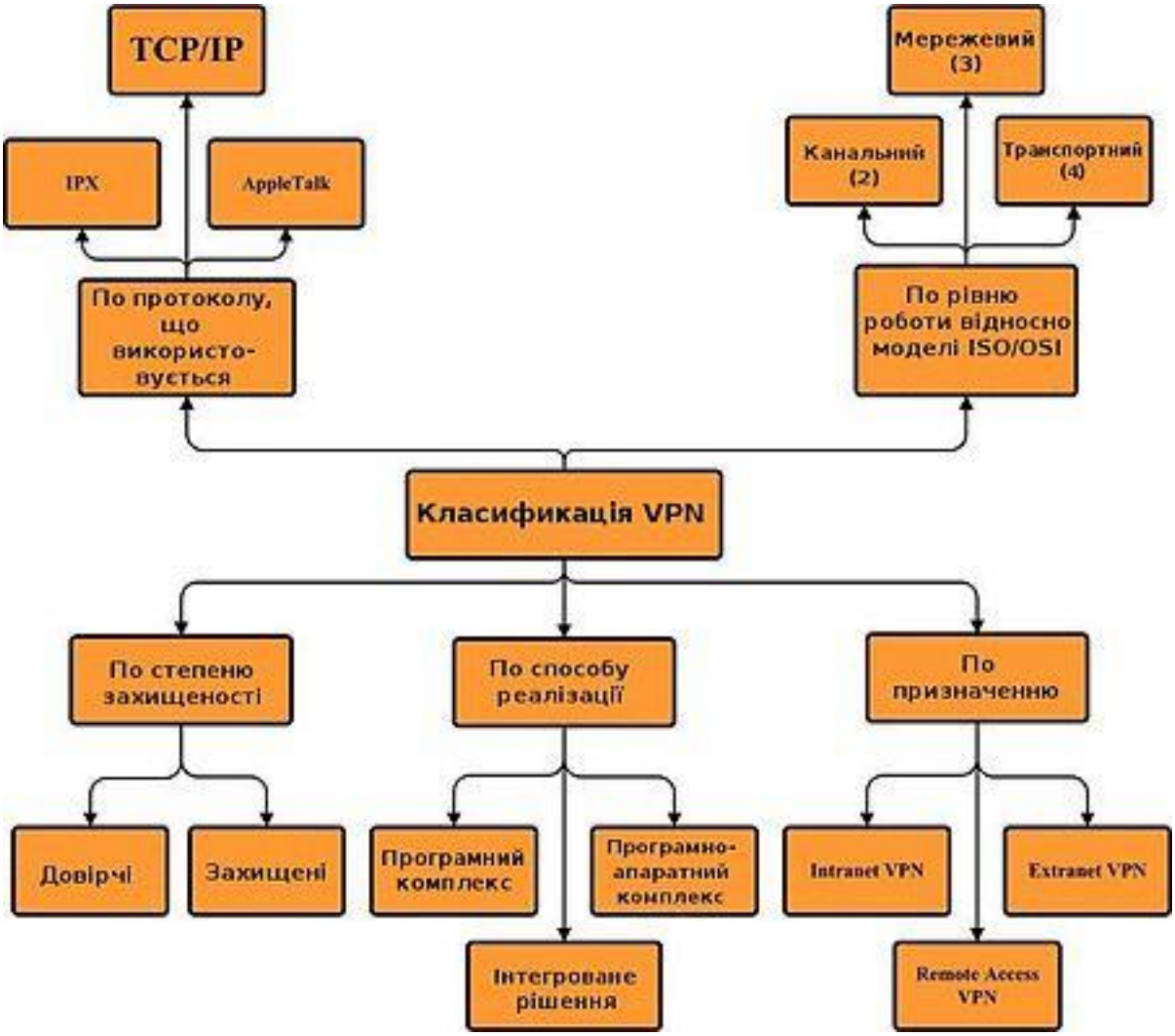

Рис 2.1. Класифікація VPN

Найчастіше для створення віртуальної мережі використовують інкапсуляцію протоколу PPP в який-небудь інший протокол — IP (такий спосіб використовує реалізація PPTP — англ. Point-to-Point Tunneling Protocol) або Ethernet (PPPoE) (хоча і вони мають відмінності). Технологію VPN останнім часом використовують не тільки для створення приватних мереж, але і деякі провайдери на пострадянському просторі для надання виходу в Інтернет.

Зазвичай, при створенні VPN, використовують під'єднання типу точка-точка до певного сервера, або установку ethernet-тунелю з певним сервером, при якій

тунелю призначають певну підмережу. Сервер VPN при цьому виконує функції маршрутизації та фільтрування трафіку для доступу до локальної мережі через VPN.

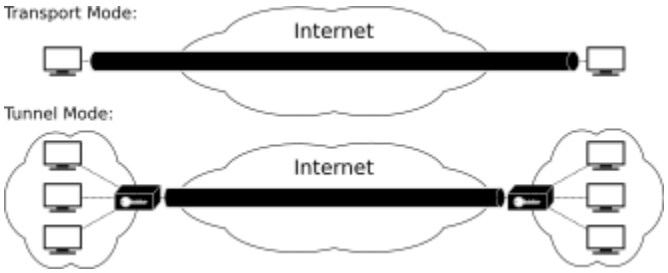

Рис 2.2 VPN - міст

За використання такого підходу ми все ще маємо можливість фільтрувати трафік через спосіб під'єднання (наприклад, використовувати для локальної мережі та для віддалених користувачів різні фільтри), але усунуто потребу налаштування маршрутизації, а віддалені машини включаються прямо в локальну мережу, бачать ресурси, навіть спроможні використовувати широкосмугові посилки взагалі без додаткового налаштування. Через такий VPN у них відображаються всі комп'ютери локальної мережі Windows, всі доступні XDMCP-сервери при XDMCP broadcast.[1]

Список найбільш поширених протоколів VPN:

1. IPSec (англ. IP security) — часто використовується поверх IPv4;
2. PPTP (англ. Point-to-point tunneling protocol) — розроблявся спільними зусиллями декількох компаній, включаючи Microsoft;
3. PPPoE або PPP (англ. Point-to-Point Protocol over Ethernet) ;
4. L2TP (англ. Layer 2 Tunnelling Protocol) — використовується в продуктах компаній Microsoft і Cisco;
5. L2TPv3 (англ. Layer 2 Tunnelling Protocol version 3);
6. OpenVPN SSL VPN з відкритим вихідним кодом, підтримує режими PPP, bridge, point-to-point, multi-client server.

PPTP протокол

PPTP означає протокол тунелювання точка-точка. Це один із найперших розроблених протоколів VPN, який широко використовувався через його простоту та легкість налаштування. PPTP створює безпечний тунель між вашим пристроєм і сервером VPN, шифруючи дані, що проходять через нього.[2]

Однак PPTP став менш популярним через деякі вразливості безпеки, виявлені з часом. Шифрування, яке використовується в PPTP, не таке надійне, як більш сучасні протоколи, що робить його потенційно вразливим до певних атак. Тому багато експертів із безпеки радять не використовувати PPTP для конфіденційних даних або там, де потрібна висока безпека.[4]

Хоча він все ще може підтримуватися деякими пристроями та системами з міркувань сумісності, інші протоколи VPN, як-от OpenVPN, L2TP/IPsec, IKEv2/IPsec тощо, зазвичай вважаються більш безпечними альтернативами PPTP. Вибираючи протокол VPN, важливо віддати пріоритет безпеці та врахувати міцність шифрування та потенційні вразливості кожного протоколу.

L2TP протокол

L2TP або Layer 2 Tunneling Protocol — ще один протокол VPN, який, як і PPTP, створює безпечний тунель для передачі даних. Однак сам L2TP не забезпечує шифрування. Щоб захистити дані, що передаються, він часто використовується в поєднанні з IPsec (Internet Protocol Security).[8]

L2TP/IPsec поєднує найкращі функції L2TP та IPsec:

1) L2TP: встановлює тунель для передачі даних;

2) IPsec: забезпечує шифрування та автентифікацію даних, що проходять через тунель.

Ця комбінація створює більш безпечне з’єднання, ніж лише L2TP, усуваючи деякі проблеми безпеки, пов’язані з такими протоколами, як PPTP. L2TP/IPsec

широко підтримується на різних пристроях і операційних системах, що робить його популярним вибором для з’єднань VPN, де безпека є пріоритетом.[1]

Однак, як і будь-яка технологія, L2TP/IPsec має свої обмеження. Деякі брандмауери можуть блокувати трафік L2TP/IPsec, і він може бути повільнішим порівняно з іншими протоколами через витрати на шифрування. Незважаючи на ці обмеження, протокол L2TP/IPsec все ще вважається досить безпечним протоколом VPN, якщо його правильно налаштувати та використовувати.[2]

З VPN розібралися, загалом це досить хороша захищена і необхідна технологія в сучасному світі. Перейдемо до другого типу підключення до віддалених ресурсів підприємства.[5]

2.3. Доступ до інформаційних ресурсів засобами RDP

RDP (англ. Remote Desktop Protocol, протокол віддаленого робочого стола) — протокол прикладного рівня, що використовується для забезпечення віддаленої роботи користувача із сервером, на котрому запущений сервіс термінальних з'єднань.

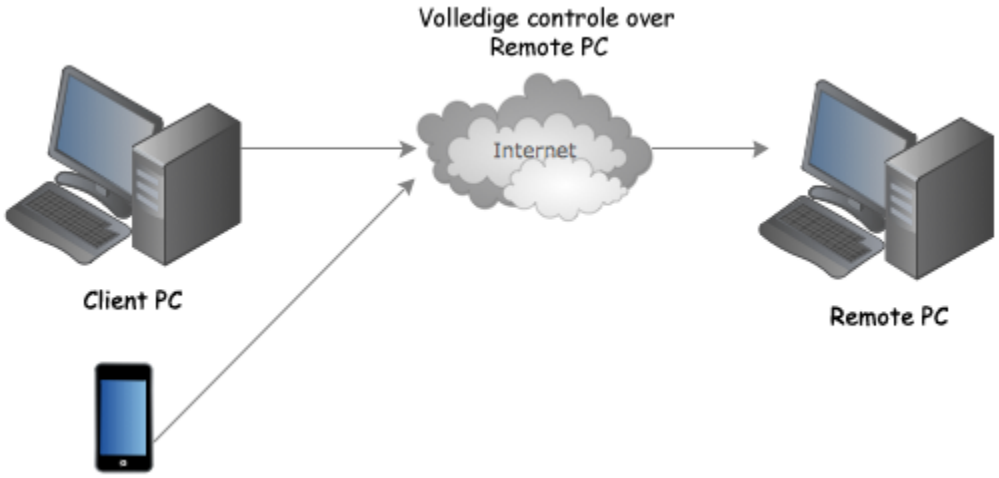

Рис 2.3. Схема RDP підключення

Клієнти існують практично для всіх версій Windows (включаючи Windows CE та Mobile), Linux, FreeBSD, Mac OS X. За умовчанням, використовується порт TCP 3389. Офіційна назва Майкрософт для клієнтського ПЗ — Remote Desktop Connection

або Terminal Services Client (TSC), зокрема, клієнт у Windows XP/2003/Vista називається mstsc.exe.[10]

Це власний протокол, розроблений корпорацією Майкрософт, який дозволяє користувачеві підключатися до іншого комп'ютера та керувати ним віддалено через мережу.

Ось як зазвичай працює RDP:

1) Віддалений доступ: на комп'ютері, яким потрібно керувати (віддаленому комп'ютері), має бути ввімкнено RDP і налаштовано приймати вхідні з'єднання RDP;

2) З'єднання: користувач іншого комп'ютера (клієнтського комп'ютера) використовує клієнтське програмне забезпечення RDP, наприклад Microsoft Remote Desktop або інші програми сторонніх розробників, щоб ініціювати підключення до віддаленого комп'ютера;

3) Автентифікація: користувач вводить облікові дані (ім'я користувача та пароль), необхідні для доступу до віддаленого комп'ютера. У разі успішної автентифікації віддалений робочий стіл іншого комп'ютера з'являється на екрані клієнта, що дозволяє їм взаємодіяти з ним так, ніби вони фізично присутні на цій машині.[5]

RDP зазвичай використовується для різних цілей:

1) Віддалена допомога: IT-персонал підтримки часто використовує RDP для віддаленого усунення несправностей і вирішення проблем на комп'ютерах користувачів без фізичної присутності;

2) Віддалена робота: широко використовується в бізнесі, щоб надати працівникам доступ до своїх робочих комп'ютерів з дому чи інших віддалених місць, забезпечуючи гнучкість робочого середовища.

Однак, хоча RDP зручний для віддаленого доступу, надзвичайно важливо забезпечити належні заходи безпеки:

Загрози безпеці: якщо протокол RDP не налаштований безпечно, він може бути вразливим до атак, таких як атаки грубої сили, які намагаються вгадати облікові дані для входу або використовують відомі вразливості в самому протоколі.[4]

Практики безпеки: передові практики включають використання надійних унікальних паролів, увімкнення автентифікації на рівні мережі (NLA), постійне оновлення програмного забезпечення RDP і використання VPN для додаткового рівня безпеки під час віддаленого доступу до комп'ютерів через Інтернет.[11]

Завжди надавайте пріоритет безпеці під час використання RDP, щоб запобігти несанкціонованому доступу або можливим порушенням.

2.4. NAT - Network Address Translation

NAT означає трансляцію мережевих адрес. Це технологія, яка використовується в маршрутизаторах, щоб дозволити кільком пристроям у локальній мережі спільно використовувати одну публічну IP-адресу для підключення до Інтернету.[2]

Ось як це працює:

1) Приватні IP-адреси: Пристроям у локальній мережі призначаються приватні IP-адреси, наприклад, у діапазонах 192.168.x.x, 10.x.x.x або від 172.16.x.x до 172.31.x.x. Ці адреси не маршрутизуються через Інтернет;

2) Загальнодоступна IP-адреса: маршрутизатор має загальнодоступну IP-адресу, яка використовується для зв'язку з пристроями в Інтернеті. Ця публічна IP-адреса призначається провайдером і є унікальною;

3) Переклад: коли пристрій у локальній мережі хоче з'єднатися з Інтернетом, маршрутизатор перетворює приватну IP-адресу пристрою на загальнодоступну IP-адресу. Вихідні пакети даних із пристрою змінюються для заміни приватної IP-адреси загальнодоступною IP-адресою маршрутизатора; [10]

4) Відображення портів: NAT також відстежує, який пристрій ініціював вихідне з'єднання та які порти використовувалися. Коли відповідь надходить з

Інтернету, маршрутизатор використовує цю інформацію, щоб надіслати вхідний пакет даних на правильний пристрій у локальній мережі.

NAT має кілька переваг:

1) Збереження IP-адрес: це дозволяє великій кількості пристроїв у локальній мережі отримувати доступ до Інтернету за допомогою однієї загальнодоступної IP-адреси, що допомагає зберегти адреси IPv4;

2) Безпека: NAT діє як основний брандмауер, приховуючи IP-адреси внутрішніх пристроїв від Інтернету, забезпечуючи певний рівень безпеки через невідомість.

Однак NAT іноді може спричиняти проблеми з певними програмами чи службами, які покладаються на пряме вхідне з'єднання з Інтернету, як-от онлайн-ігри чи однорангові мережі. У таких випадках можна використовувати переадресацію портів або UPnP (Universal Plug and Play), щоб відкрити певні порти та дозволити вхідному трафіку досягти певного пристрою в локальній мережі.[12]

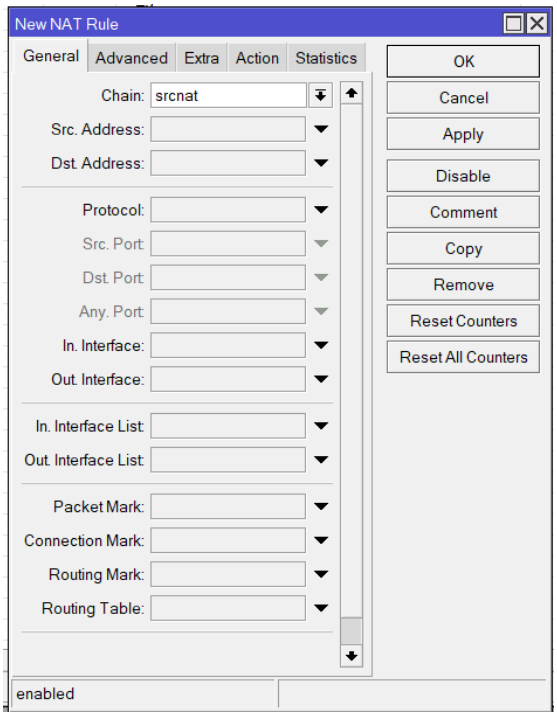

Рис 2.4. Правило NAT microtik вкладка загальні

Перетворення адреси методом NAT може відбуватися майже будь-яким маршрутизуючим пристроєм — маршрутизатором[9], сервером доступу, міжмережевим екраном (фаєрволом). Найбільш популярним є SNAT, суть механізму котрого складається у заміні адреси джерела (англ. source) при проходженні пакету в одну сторону і зворотній заміні адреси призначення (англ. destination) у зворотному пакеті. Поряд з адресами джерело/призначення можуть також замінюватися номера портів джерела і призначення.[12]

Приймаючи пакет від локального комп'ютера, роутер переглядає IP-адресу призначення. Якщо це локальна адреса, то пакет пересилається іншому локальному комп'ютерові. Якщо ні, то пакет слід переслати назовні до інтернету. Але зворотною адресою у пакеті вказана локальна адреса комп'ютера, котра з інтернету буде недоступна. Тому роутер «на льоту» транслює (підміняє) зворотню IP-адресу пакету на свою зовнішню (видиму з інтернету) IP-адресу, а також міняє номер порту (щоб розрізняти зворотні пакети, адресовані різним локальним комп'ютерам). Комбінацію, потрібну для зворотної підстановки, роутер зберігає у себе у тимчасовій таблиці. Через деякий час після того, як клієнт і сервер закінчать обмінюватися пакетами, роутер зітре у себе в таблиці запис про n-м порт за строком давнини.[11]

Окрім source NAT (надання користувачам локальної мережі з внутрішніми адресами доступу до мережі Інтернет) часто застосовується також destination NAT, коли трафік ззовні транслюється міжмережевим екраном на комп'ютер користувача у локальній мережі, котрий має внутрішню адресу і тому недоступний ззовні мережі безпосередньо (без NAT).[10]

Існує 3 базових концепції трансляції адрес: статична (Static Network Address Translation), динамічна (Dynamic Address Translation), перевантажена (NAPT, NAT Overload, PAT).

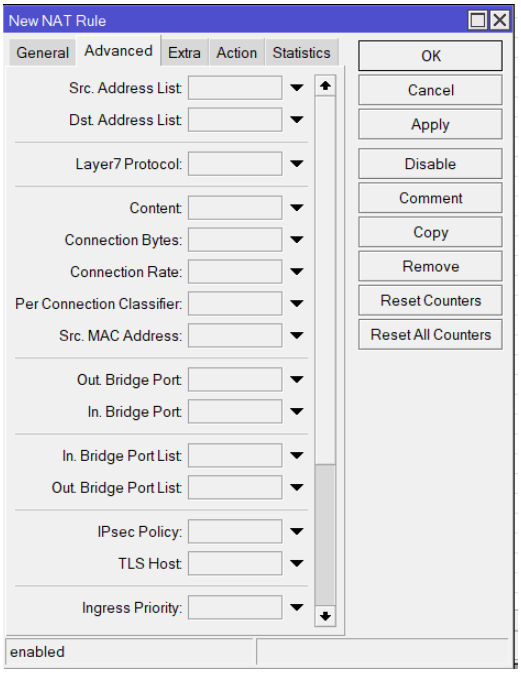

Рис 2.5. Правило NAT microtik вкладка Advanced

Статичний NAT — відображення незареєстрованої IP-адреси на зареєстровану IP-адресу на основі один до одного. Особливо корисно, коли пристрій повинен бути доступним зовні мережі.[3]

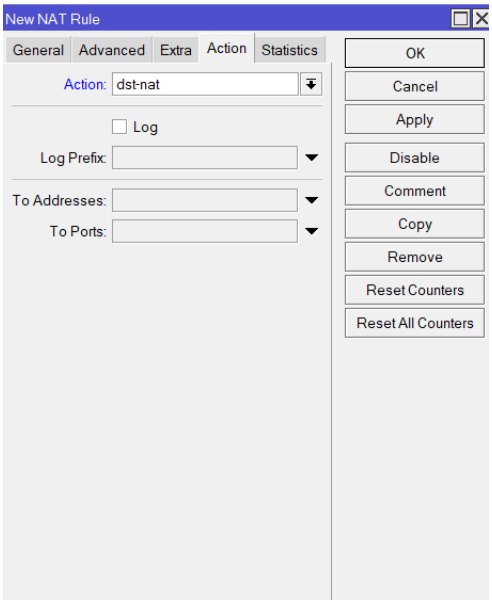

Рис 2.6. Правило NAT microtik вкладка Action

Динамічний NAT — відображує незареєстровану IP-адресу на зареєстровану адресу з групи зареєстрованих IP-адрес. Динамічний NAT також встановлює безпосереднє відображення між незареєстрованими і зареєстрованими адресами, але відображення може мінятись в залежності від зареєстрованої адреси, доступної у купі адрес, під час комунікації.[2]

Перевантажений NAT (NAPT, NAT Overload, PAT, маскарадинг) — форма динамічного NAT, котрий перетворює декілька незареєстрованих адрес у єдину зареєстровану IP-адресу, використовуючи різноманітні порти. Відомий також як PAT (Port Address Translation). При перевантаженні кожен комп'ютер у приватній мережі транслюється у ту ж саму адресу, але з різним номером порту.[4]

Механізм NAT визначений у RFC 1631, RFC 3022.

2.5. Типи NAT

Класифікація NAT, часто зустрічається у зв'язку з VoIP.[2] Термін «сполука» використаний у значенні «послідовний обмін пакунками UDP».

Симетричний NAT (Symmetric NAT) — трансляція, при якій кожне сполучення, ініційоване парою «внутрішня адреса: внутрішній порт» перетворюється у вільну унікальну, випадково вибрану пару «публічну адресу: публічний порт». При цьому, ініціація сполуки з публічної мережі неможлива.[5]

Cone NAT, Full Cone NAT — однозначна (взаємна) трансляція між парами «внутрішня адреса: внутрішній порт» і «публічна адреса: публічний порт». Кожен зовнішній хост може ініціювати сполуку з внутрішнім хостом (якщо це дозволене у правилах міжмережевого екрана (брандмауера)).

Address-Restricted cone NAT, Restricted cone NAT — постійна трансляція між парою «внутрішня адреса: внутрішній порт» і «публічна адреса: публічний порт». Кожна сполука, ініційована з внутрішньої адреси, дозволяє надалі отримувати йому пакунки з будь-якого порту того публічного хоста, до якого він відправляв пакунок(ки) раніше.[10]

Port-Restricted cone NAT — трансляція між парою «внутрішня адреса: внутрішній порт» і «публічна адреса: публічний порт», при котрої пакунки що надходять до внутрішнього хосту тільки з одного порту публічного хоста — того, на котрий внутрішній хост вже відправляв пакунок.[11]

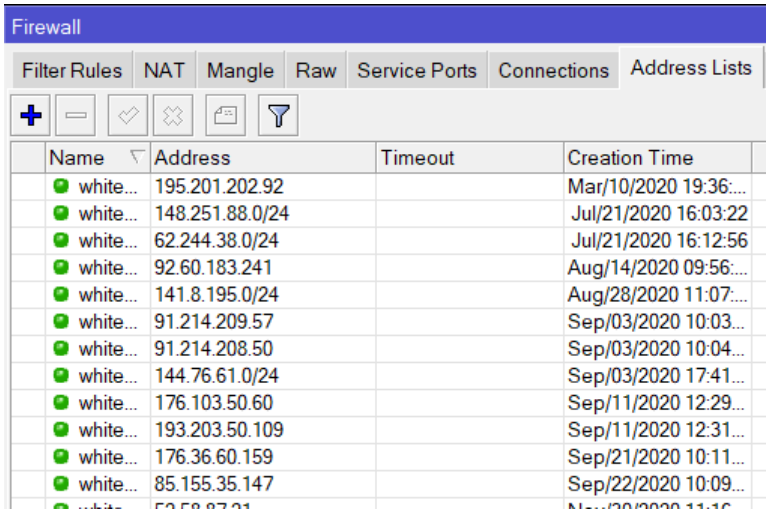

Рис 2.7. Білий список microtik

В сучасних маршрутизаторах існують так звані правила брандмауру та білі листи, завдяки цим правилам ми можемо налаштувати будь який зовнішній порт для підключення локального ресурсу, а завдяки білим спискам обмежети доступ по IP адресі користувачу.[12]

2.6. Microtik

Як ви вже могли помітити в рисунках я приводив приклади на правила та налаштування саме даної фірми пристрою, в подальшому саме з цієї марки марштуризатором буде співпрацювати наш програмний модуль, а тепер коротко про дану компанію.

MikroTik — латвійська компанія, відома своїм мережевим обладнанням і програмним забезпеченням. Вони виробляють широкий спектр маршрутизаторів, комутаторів та інших мережевих пристроїв, включаючи бездротові системи та програмне забезпечення для керування мережею. Продукти MikroTik популярні

завдяки своїй універсальності, доступності та надійності, особливо в мережах малого та середнього розміру.

Основні аспекти пристроїв і програмного забезпечення MikroTik включають:

1) RouterOS: операційна система MikroTik для своїх пристроїв називається RouterOS. Це операційна система на базі Linux, яка працює на їхніх маршрутизаторах і комутаторах, надаючи широкий набір функцій для керування мережею, включаючи маршрутизацію, брандмауер, керування бездротовими точками доступу, функції VPN тощо;

2) Універсальність: пристрої MikroTik відомі своєю гнучкістю та багатими можливостями. Вони відповідають різноманітним мережевим потребам, від простих домашніх налаштувань до складних корпоративних середовищ;

3) Доступність: продукти MikroTik часто є економічно ефективнішими порівняно з подібним мережевим обладнанням інших виробників. Це робить їх популярним вибором, особливо для невеликих підприємств або окремих осіб, які шукають надійні мережеві рішення, не розбиваючи гроші;

4) Налаштування: RouterOS дозволяє широке налаштування та конфігурацію, надаючи мережевим адміністраторам детальний контроль над своїми мережами. Він підтримує різні протоколи та має інтерфейс командного рядка (CLI) для досвідчених користувачів.

Однак завдяки потужним функціям і гнучкості, які надають пристрої MikroTik, вони можуть мати крутішу криву навчання для тих, хто тільки починає працювати в мережі або адмініструвати мережу. Рекомендується добре розуміти мережеві концепції, перш ніж занурюватися в розширені функції пристроїв MikroTik і RouterOS.

Їхні пристрої використовуються по всьому світу в різних сценаріях, від домашніх мереж до інтернет-провайдерів, компаній і навіть у сільській місцевості та регіонах, що розвиваються, завдяки своїй доступності та надійності.

2.7. NAS

NAS означає Network Attached Storage. Це стосується запам’ятовуючого пристрою або системи, яка призначена для надання послуг зберігання та обміну файлами багатьом користувачам і клієнтським пристроям через мережу, як правило, локальну мережу (LAN) або Інтернет.

Основні характеристики пристрою NAS включають:

1) Зберігання файлів: пристрої NAS призначені в основному для зберігання та керування файлами, документами, мультимедійним вмістом та іншими даними. Вони часто містять кілька жорстких дисків, налаштованих у різних конфігураціях RAID (надлишкового масиву незалежних дисків) для резервування даних, продуктивності або балансу обох;

2) Доступність мережі: пристрої NAS доступні через мережу, що дозволяє кільком користувачам або пристроям отримувати доступ до збережених даних одночасно. Цей доступ можна забезпечити за допомогою різних протоколів, таких як SMB/CIFS (використовується Windows), NFS (використовується системами на базі Unix), FTP, HTTP або спеціалізовані власні протоколи;

3) Резервне копіювання та захист даних: багато систем NAS пропонують вбудовані функції резервного копіювання, що дозволяє створювати резервні копії за розкладом або синхронізувати з іншими пристроями чи хмарними службами зберігання. Крім того, конфігурації RAID у пристроях NAS можуть забезпечити резервування даних, захищаючи від збоїв дисків;

4) Потокове передавання медіа: деякі пристрої NAS оснащені можливостями медіа-сервера, що дозволяє користувачам передавати мультимедійний вміст, наприклад відео, музику та фотографії, на такі пристрої, як смарт-телевізори, медіаплеєри або мобільні пристрої в мережі;

5) Віддалений доступ: багато сучасних систем NAS забезпечують функції віддаленого доступу, що дозволяє користувачам отримувати доступ до файлів, що

зберігаються на NAS, поза межами локальної мережі. Цьому часто сприяють безпечні протоколи або спеціалізовані програми, надані виробником NAS.

Пристрої NAS зазвичай використовуються вдома, на малих підприємствах і підприємствах для централізації та керування даними, обміну файлами між кількома користувачами та забезпечення резервного копіювання даних і безпеки. Вони бувають різних розмірів і місткості, починаючи від невеликих одиночних накопичувачів і закінчуючи більшими системами з кількома відсіками, розробленими для великих потреб у сховищах.

2.8. RDS

Термінальний сервер, також відомий як Remote Desktop Services (RDS) у новіших версіях Windows, — це серверна технологія в операційній системі Microsoft Windows, яка дозволяє кільком користувачам отримувати доступ і використовувати програми або повне середовище робочого столу віддаленого сервера. через мережу.

Ось як зазвичай працюють термінальний сервер або служби віддаленого робочого стола:

1) Конфігурація сервера: Сервер Windows, на якому запущені служби терміналів або служби віддаленого робочого столу, налаштовано та налаштовано на можливість кількох одночасних віддалених підключень;

2) Доступ користувача: користувачі можуть підключатися до сервера терміналів зі своїх власних пристроїв (ПК з Windows, Mac, планшетів або смартфонів) за допомогою протоколу віддаленого робочого стола (RDP). Вони вводять IP-адресу сервера або доменне ім’я разом зі своїми обліковими даними для встановлення віддаленого з’єднання;

3) Робота з віддаленим робочим столом: після підключення користувачі можуть отримати доступ або до повного робочого середовища, або до певних програм, розміщених на сервері, залежно від конфігурації. Кожен користувач

отримує власний сеанс, ізольований від інших, що дозволяє йому працювати незалежно.

Основні функції та переваги сервера терміналів або служб віддаленого робочого стола включають:

- Централізоване керування додатками: додатки встановлюються та керуються на сервері, що зменшує необхідність індивідуальних установок на пристрої кожного користувача;
- Спільне використання ресурсів: користувачі можуть спільно використовувати такі ресурси, як файли або принтери, доступні на сервері;
- Віддалена робота: полегшує сценарії віддаленої роботи, дозволяючи користувачам отримувати доступ до свого робочого середовища з будь-якого місця, де є підключення до Інтернету;
- Централізоване адміністрування: IT-адміністратори можуть централізовано керувати доступом користувачів, програмами, налаштуваннями безпеки та оновленнями на сервері.

Сервер терміналів або служби віддаленого робочого стола зазвичай використовуються в бізнес-середовищах, де кільком користувачам потрібен доступ до певних програм або стандартизованого робочого середовища. Він спрощує адміністрування, підвищує безпеку за рахунок централізації даних і забезпечує гнучкі сценарії роботи, надаючи віддалений доступ до узгодженого обчислювального середовища.

2.9. MSSQL

SQL означає мову структурованих запитів. Це мова програмування, призначена для керування реляційними базами даних і маніпулювання ними. SQL дозволяє вам виконувати такі завдання, як отримання даних із бази даних, вставка та оновлення даних, створення та зміна схем бази даних (таблиці, індекси тощо), а також керування дозволами в системі баз даних. Він широко використовується в

різних системах баз даних, таких як MySQL, PostgreSQL, SQLite, Microsoft SQL Server і багатьох інших.[16]

MS SQL, скорочення від Microsoft SQL Server, — це реляційна система керування базами даних, розроблена Microsoft. Це надійна та багатофункціональна платформа, яка використовується для зберігання та отримання даних за запитом інших програмних програм.[17]

Він підтримує SQL як мову запитів і надає інструменти для керування базами даних, налаштування безпеки, створення таблиць і керування ними, а також оптимізації запитів для кращої продуктивності. MS SQL Server зазвичай використовується в корпоративних середовищах через його масштабованість, надійність та інтеграцію з іншими продуктами та технологіями Microsoft.[16]

Microsoft SQL Server (MSSQL) працює як система керування реляційною базою даних (RDBMS), яка структуровано зберігає дані та керує ними.

Ось огляд того, як це працює:

1) Зберігання: MSSQL організовує дані в бази даних, які містять таблиці. Таблиці мають рядки (записи) і стовпці (поля), де дані зберігаються в структурованому форматі;

2) Маніпулювання даними: користувачі взаємодіють із MSSQL за допомогою SQL (мова структурованих запитів) для виконання таких операцій, як запит, вставка, оновлення та видалення даних. Команди SQL дозволяють користувачам отримувати певні дані, змінювати існуючі дані або додавати нові дані до бази даних;

3) Контроль паралелізму: MSSQL керує декількома користувачами, які одночасно отримують доступ до бази даних за допомогою техніки, яка називається керуванням паралелізмом. Це гарантує, що транзакції ізольовані одна від одної, щоб запобігти перешкодам і зберегти цілісність даних;

4) Безпека: включає такі функції безпеки, як механізми автентифікації та авторизації для контролю доступу до бази даних, гарантуючи, що лише авторизовані користувачі можуть переглядати або змінювати дані;

5) Резервне копіювання та відновлення: MSSQL надає інструменти та функції для створення резервних копій баз даних. Це вкрай важливо для аварійного відновлення та забезпечення доступності даних навіть у разі збою системи;

6) Масштабованість і продуктивність: MSSQL розроблений для обробки великих обсягів даних і може масштабуватися для збільшення обсягів даних. Він також включає такі функції оптимізації, як індекси, оптимізація запитів і кешування для підвищення продуктивності;

7) Інтеграція: Microsoft SQL Server інтегрується з іншими продуктами та технологіями Microsoft, сприяючи безперебійному обміну даними та інтеграції з програмами, інструментами звітності та платформами розробки;

8) Адміністрування та моніторинг: MSSQL пропонує інструменти адміністрування та можливості моніторингу для керування конфігураціями сервера, налаштування продуктивності та моніторингу стану бази даних для забезпечення оптимальної роботи.

По суті, MSSQL структуровано керує даними, надає інструменти для взаємодії з цими даними, забезпечує їх безпеку та пропонує функції масштабованості та продуктивності для обробки різних робочих навантажень і розмірів даних.[17]

2.10 CRM

CRM означає управління взаємовідносинами з клієнтами. Це технологія та стратегія, які використовують компанії для керування взаємодією та стосунками з поточними та потенційними клієнтами. Система CRM дозволяє організаціям оптимізувати процеси, покращити обслуговування клієнтів, збільшити продажі та сприяти маркетинговим зусиллям, відстежуючи взаємодію з клієнтами, потенційні клієнти, інформацію про клієнтів тощо.[14]

Основні аспекти системи CRM включають:

1) Централізована база даних: зберігання даних клієнтів в одному доступному місці, включаючи контактну інформацію, історію покупок, історію спілкування та налаштування;

2) Відстеження взаємодії з клієнтом: запис взаємодії через різні канали, такі як електронні листи, телефонні дзвінки, соціальні мережі та особисті зустрічі, що дозволяє покращити спілкування та подальші дії;

3) Управління продажами та потенційними клієнтами: управління продажами, відстеження потенційних клієнтів і автоматизація процесів продажів для підвищення ефективності та коефіцієнтів конверсії;

4) Автоматизація маркетингу: цільові маркетингові кампанії, маркетинг електронною поштою та персоналізовані повідомлення на основі даних і поведінки клієнтів.

5) Підтримка та обслуговування клієнтів: забезпечення кращого обслуговування клієнтів шляхом відстеження проблем, автоматизації процесів підтримки та забезпечення своєчасних відповідей на запити клієнтів;

6) Аналітика та звітність: створення звітів і статистичних даних на основі даних клієнтів для прийняття обґрунтованих бізнес-рішень, визначення тенденцій і прогнозування майбутніх продажів або поведінки клієнтів.

Системи CRM бувають різних форм, від простих хмарних рішень до складного програмного забезпечення корпоративного рівня. Серед прикладів – Salesforce, HubSpot, Microsoft Dynamics 365 і Zoho CRM. Ці системи можна налаштовувати відповідно до конкретних потреб і робочих процесів різних компаній, допомагаючи їм ефективно керувати та підтримувати відносини з клієнтами протягом усього життєвого циклу клієнта.[15]

Висновок до другого розділу:

Досліджена інформаційна політика та способи реалізації доступу до інформаційних ресурсів. Зроблені висновки який зі способів кращий в тому чи іншому аспекті захищеності та зручності.

В плані налаштувань звичайно простіший та доступніший варіант з RDP, в плані захищенності VPN проте кожна компанія вибирає свій спосіб захисту для віддаленого підключення співробітника, а також ті налаштування які більш за все підходять даній компанії як в плані бюджетного проектування так і в плані технічної реалізації.

Принцип безпеки/Технологія	VPN	RDP(port)
Доступність	-	+
Цілісність	+	-
Конфіденційність	+	-

Табл 1. Порівняння принципів безпеки та технологій

І трішки опишу таблицю порівняльну чому по результатам аналізу ми отримуємо такі результати. Що стосується доступності, на мій погляд RDP через зовнішний порт більш доступна технологія ніж VPN та локальний айпі RDP. Досить багато провайдерів блокують можливість використання тих чи інших VPN протоколів.

Що стосується цілісності та конфіденційності тут вже навпаки, VPN більш захищений метод. В той час як в RDP можна перевірити відкритість тих чи інших портів, спробувати атакувати їх, а також це потребує більш детальних налаштувань білих листів та доступів до тих чи інших пристроїв компанії.

Розділ 3. Розробка програмного модуля керування доступом до інформаційний ресурсів.

3.1. Актуальність даної розробки.

В нашому світі доступ до інформаційних ресурсів є ключовим для створення та існування компаній. Ресурси використовуються як бухгалтерами так і звичайними працівниками. Це можу бути звичайне файлове сховище, а може бути і повноцінний термінальний сервер, в звязку з цим є необхідність забезпечити повноцінний доступ до таких або схожих ресурсів на постійній основі. Проте не завжди це виходить зробити.

На даний момент використовуються зазвичай дві технології доступу до таких ресурсів: VPN та віддалений порт. Про дані технології було описано в розділі 1.

3.2. Розробка та робота програмного модуля керування доступом до інформаційних ресурсів.

Уявімо собі ситуацію, у нас є велика компанія, в ній є айті відділі, є працівники які працюють віддалено, враховуючи наші реалії цілком звичайна справа. Але айті відділ працює по графіку, наприклад з 9 -18, а віддаленим працівникам необхідно мати доступ до ресурсів після цих годин, переїхав, відрядження – це все досить знайому, використовуючи технологію VPN проблем може і не виникнути, вже залежить від провайдера, чи є в нього якісь особливі налаштування, чи блокує він порт для тих чи інших технологій VPN. Бо на маршрутизаторі можуть бути різні налаштування.

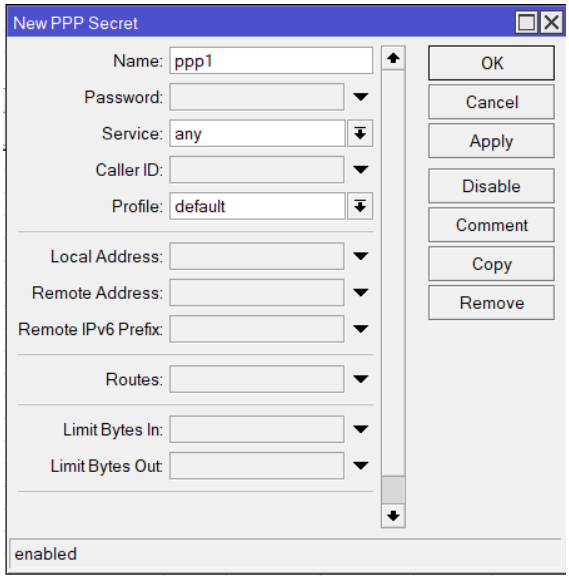

Рис 3.1. Вікно налаштувань vpn на маршрутизаторі microtik

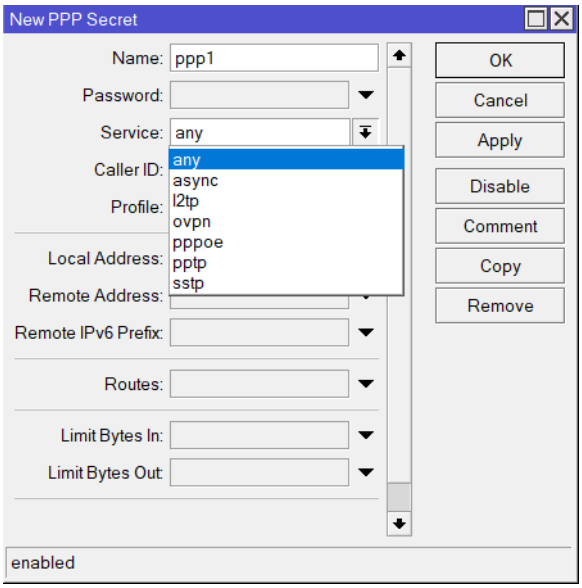

Рис 3.2. Вибір типу vpn

За основу ми будемо використовувати маршрутизатор microtik даний пристрій досить популярний і не такий дорогий як cisco, саме тому він використовується в багатьох фірмах, а його гнучкість в налаштуваннях тільки додає йому балів. На цих двох малюнках ми бачимо інтерфейс створення VPN і саме в розділі сервісів ми можемо вибрати необхідний тип VPN підключення.

Але даний тип ми не будемо використовувати тому що повернемося до простіших технологій, а саме віддалений доступ через зовнішній айпі та порт. Ось так це налаштування виглядає в середині маршрутизатора microtik.

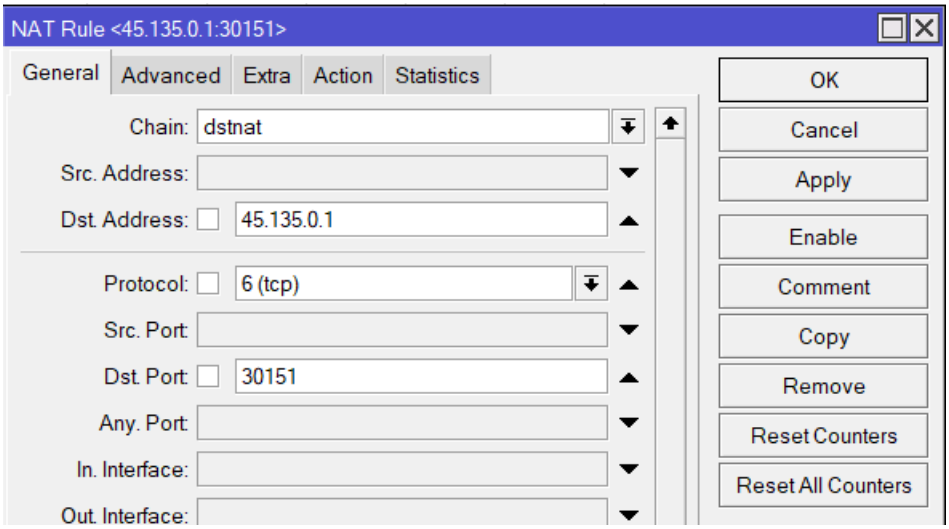

Рис 3.3. Вікно налаштування зовнішнього порта

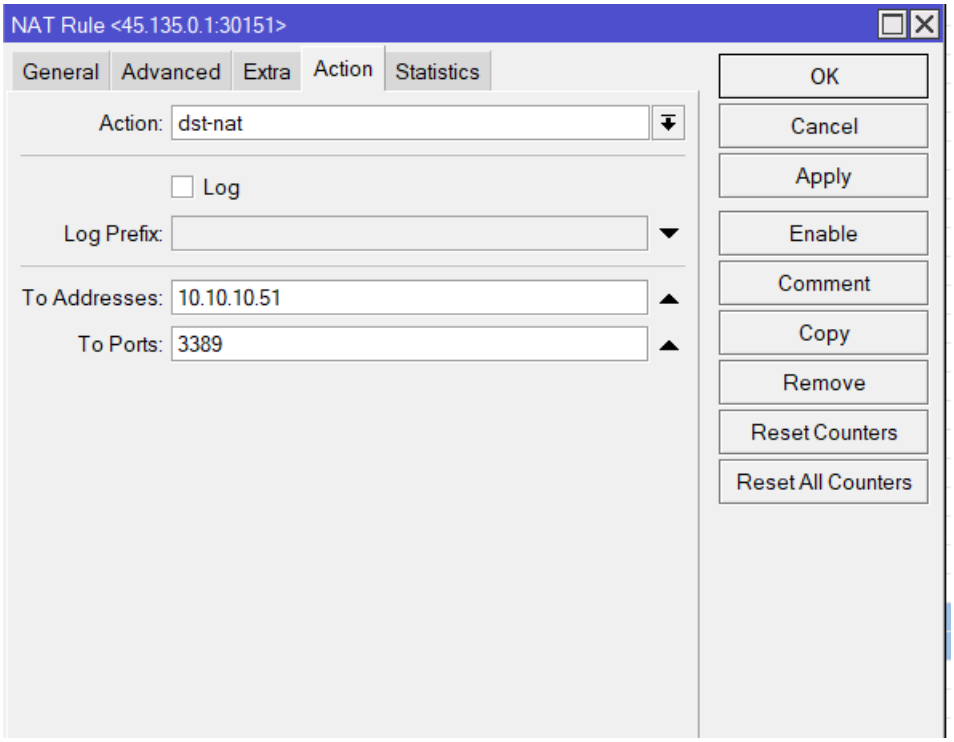

Рис. 3.4 Вікно налаштування внутрішнього порта

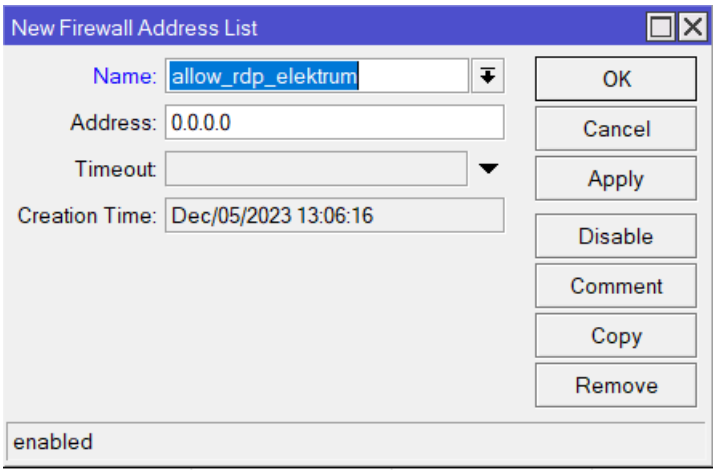

Рис 3.5 Додавання айпі в список доступу

Повертаючись до нашої компанії – ось я віддалений працівник змінив місце розташування відповідно в мене інший зовнішній айпі і мені необхідно терміново ввести якусь інформацію в 1с або в медок або відправити звіти, досить знайома ситуація, щось треба зробити терміново а віддаленого доступу немає. Чому немає? Тому що використовуючи віддалений порт, на маршрутизаторі створені правила для підключення на цей порт для цього ресурсу - правила доступу, білий лист. І в цьому білому листі немає нашого нового айпі адресу, таким чином маршрутизатор унеможливлює підключення з незнайомих айпі адрес до ресурсу і допомагає захищати інформацію від шахраїв. На годиннику 12 година ночі і турбувати айті відділ не завжди хочеться, все ж таки всі ми люди, а коли ти знаходишся у відряджені і досить часто змінюєш місце розташування а з ним і айпі адресу. В такому випадку що робити? Для цих ситуацій наша компанія і розробила такий додаток. Він використовується у внутрішніх цілях на даний момент і активно тестується та використовується нашими користувачами.

Нажаль згідно корпоративних правил я змушений змінити частину даних айпі на наступних скріншотах проте якщо ви відвідаєте нашу розробку зможете онлайн переконатися в тому, що вона працює і працює коректно.

Отже наш програмний модуль має власний веб інтерфейс, відповідно на наступних скрішотах я більш детально розповім що і як.

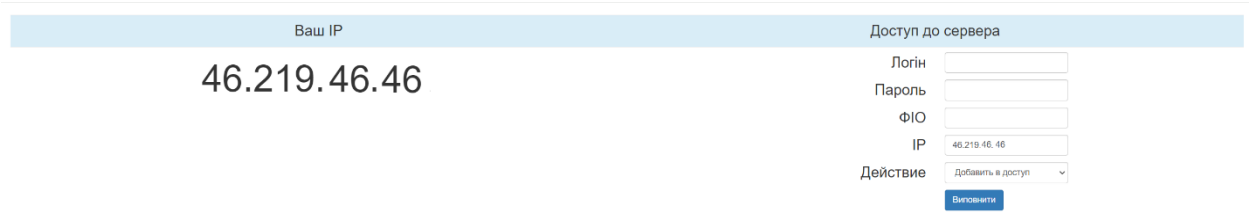

Рис 3.6. Веб інтерфейс додатку

На головній сторінці вже додана необхідна нам інформація, а саме наш айпі. У кожного провайдера є свої айпі межі, так як доступ до ресурсів забезпечується додаванням айпі в білий список то для зручності ця інформація виведена на екран, завдяки інтеграції з сайтом ipwhois. Також ми бачимо що дана адреса автоматично вноситься в панель зправа, тобто нам залишається лише ввести дані логіну та паролю до маршрутизатора, вибрати дію та бажано ввести коментарій в розділі ФІО. Дана програма розроблялася до повномасштабного вторгнення та як внутрішній тестовий зразок тому нажаль не є українізованою. В майбутньому ми переведемо даний інтерфейс повністю на українську версію.

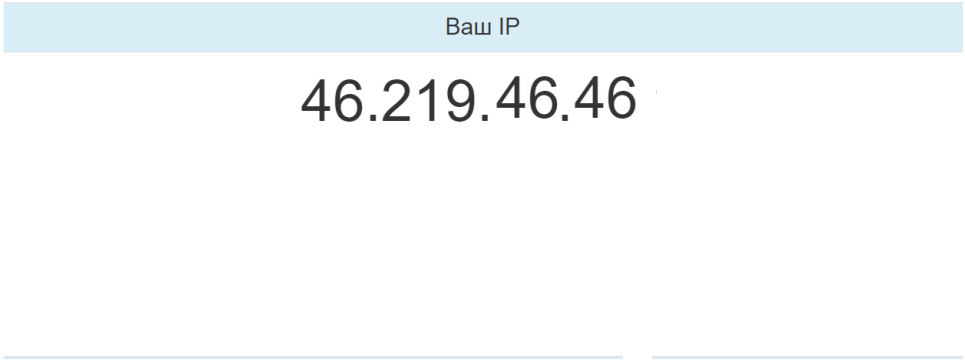

Рис 3.7. Ваш поточній зовнішній айпі

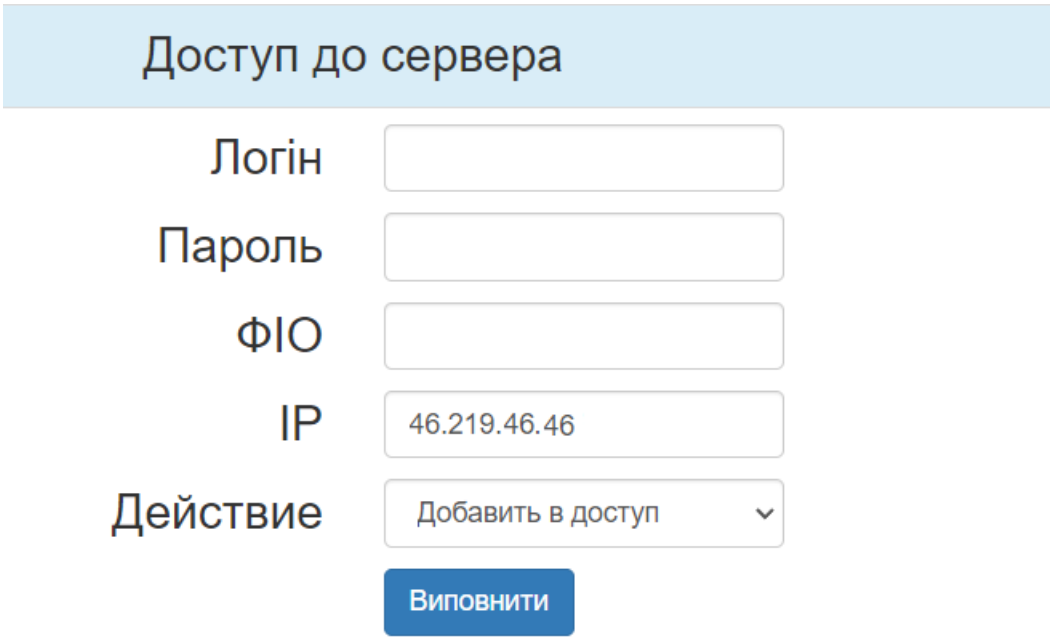

Рис 3.8. Панель введення даних

Що стосується логіну та паролю, де необхідно брати ці дані, що це за список? Даний список створюється на самому маршрутизаторі, таким чином користувач може самостійно додавати свій айпі. Звичайно необхідно слідкувати за актуальністю цих даних, так як люди наймаються на роботу на звільняються, цим вже займається безпосередньо сама компанія яка орендує в нас сервер, в поштовому варіанті, тільки з перевіреного емейлу приймаються запити на додавання або видалення акантів зі списків доступу, зазвичай це обліковий запис пошти керівника відділу чи керівника компанії.

Більш детально розглянемо це пізніше, а зараз прошу звернути увагу на пункт дія, там є не тільки додавання айпі адреси, а і видалення існуючої. Взагалі даний пункт це не просто про безпеку, це вже про освіченість користувача та його самосвідомість. В нашому світі необхідні курси по захисту інформації для кожного працівника бо ті чи інші дії можуть призвести до витоку інформації, а це в свою чергу до збитків компанії. І важливо прибирати за собою: що я маю на увазі,

користувач у відряджені постійно змінює айпі адресу, типовий випадок, бажано щоб він не тільки додав нову адресу, а і видаляв стару, таким чином підчищаючи за собою вікна можливості для атак на інформаційну структуру компанії.

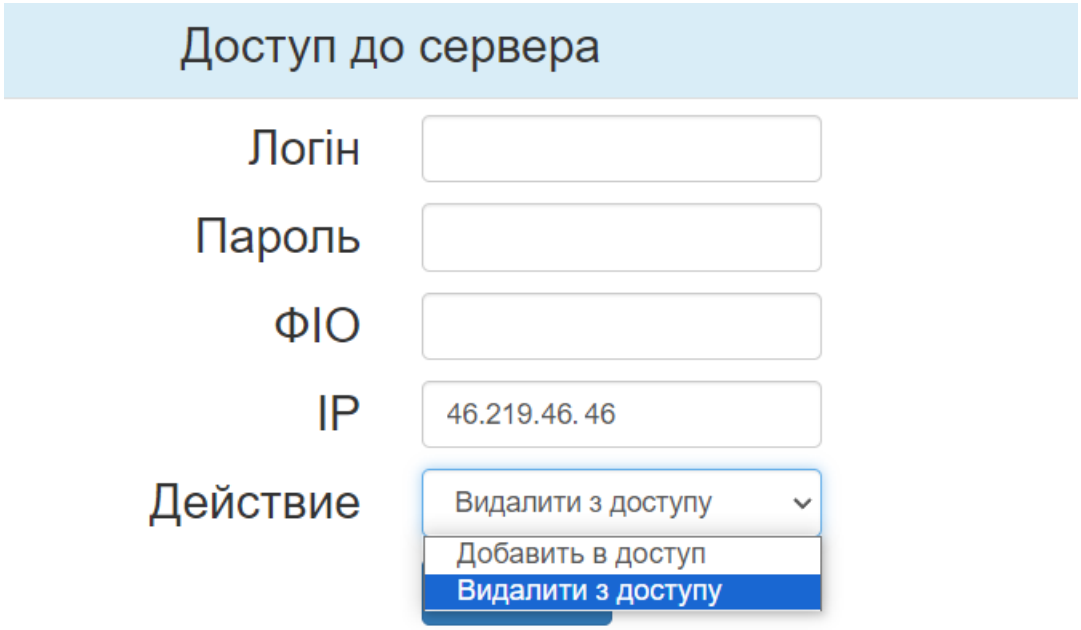

Рис 3.9. Кнопка вибору дії

А тепер перейдемо до взаємодії даного програмного модуля з іншим програмним забезпеченням.

На сервері 10.10.10.230 розгорнута база SQL відповідна таблиця в яку вносятся логіни та паролі на ті чи інші мікротіки та маршрутизатори за допомогою яких ми і можемо додавати айпі.

Ось як це виглядає в коді програми:

```
$magic_db = '*******';
$magic_db_user = '******';
$magic_db_password = '*********';
$con1 = mysql_connect ('10.10.10.230', $magic_db_user, $magic_db_password);
mysql_select_db($magic_db, $con1);
mysql_query ("SET CHARACTER SET utf8", $con1);
```

```
mysql_query ("set names utf8");
$_SERVER['REMOTE_ADDR'] = $_SERVER['HTTP_CF_CONNECTING_IP'];
$ip = $_SERVER['REMOTE_ADDR'];
if (isset($_REQUEST['user']) || isset($_REQUEST['pass']))
```

Дана таблиця виведена в нашу crm систему, що допомагає швидко редагувати її та вносити зміни. Дані нажаль мені доводиться замалювати так як це конфіденційна програма. Доступ в саму crm мають тільки довірені користувачі.

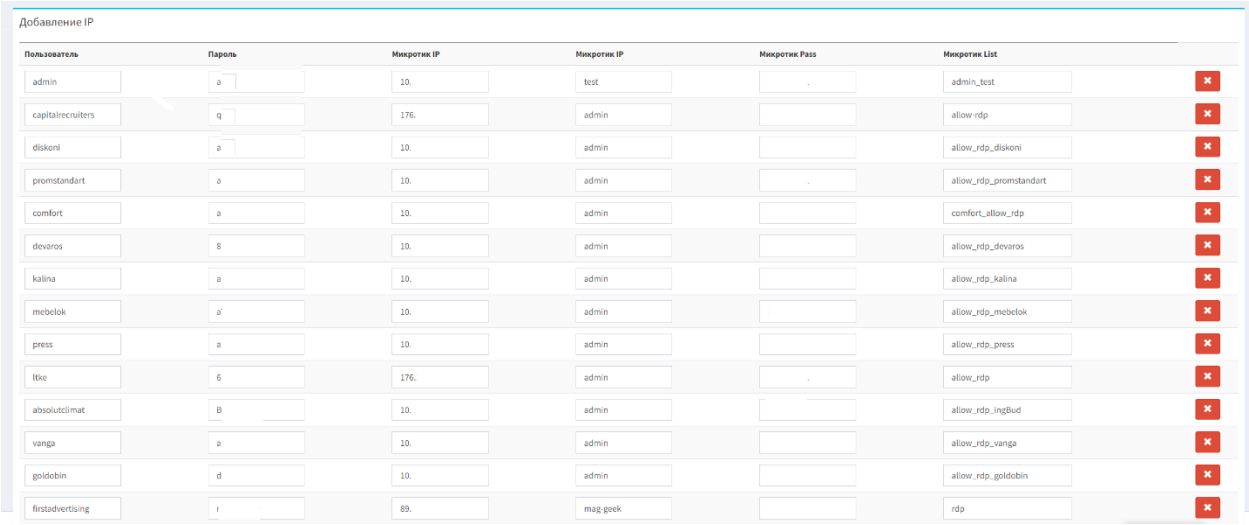

Рис 3.10 Вигляд списку доступів в crm системі

На даному рисунку ми бачимо такі позиції як логін, пароль, айпі адреса мікротику, логін та пароль до мікротіка на назву білого листа, відповідно в який потім потрапляють айпі які ми додаємо.

Таким чином даний програмний модуль функціонує одразу ще з двома системами – crm системою, а також базою даних sql, відправляючі дані та запити до них та отримуючи відповіді, сам процес виглядає такми чином, ви вводите дані, вони потрапляють в таблицю даних, а звідти вже в маршрутизатор.

https://2ip.supportio.ua/ - посилання на сайт та доступ до додатку.

РОЗДІЛ 4 (5). ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

4.1. Заповідні території в Україні

Природний заповідник – одна з категорій природно-заповідного фонду України (колишня назва – “державний заповідник”). Це установа, що має природоохоронний і науково-дослідний статус загальнодержавного значення.

Природні заповідники створюються і діють для збереження в первинному стані природних комплексів та об’єктів певних територій, дослідження природних процесів усередині них, оцінювання впливу та змін навколишнього природного середовища, сталого використання природних ресурсів тощо.

В Україні нараховується 19 природних заповідників, що охоплюють території 12 областей та АР Крим. Найбільша кількість природних заповідників розташована в Криму.

Перелік природних заповідників

Назва	Область
Ґорґани	Івано-Франківська
Дніпровсько-Орільський	Дніпропетровська
Древлянський	Житомирська
Єланецький степ	Миколаївська
Казантипський	АР Крим
Канівський	Черкаська
Карадазький	АР Крим
Кримський	АР Крим
Луганський	Луганський
Медобори	Тернопільська
Мис Мартьян	АР Крим
Михайлівська цілина	Сумська

Опуцький	АР Крим
Поліський	Житомирська
Розточчя	Львівська
Рівненська	Рівненська
Черемський	Волинська
Український степовий	Донецька
Ялтинський гірсько-лісовий	АР Крим

Заповідна зона є найбільш важливою на території парку, оскільки саме вона містить в собі найбільш цінні природні комплекси. Згідно Закону України “Про природно-заповідний фонд України” на цю зону поширюється правовий режим, встановлений для заповідників.

Як відомо, природні заповідники - природоохоронні, науково-дослідні установи загальнодержавного значення, що створюються з метою збереження в природному стані типових або унікальних для даної ландшафтної зони природних комплексів з усією сукупністю їх компонентів, вивчення природних процесів і явищ, що відбуваються в них, розробки наукових засад охорони навколишнього природного середовища, ефективного використання природних ресурсів та екологічної безпеки (ст.15 Закону «Про природно-заповідний фонд України»).

Основними завданнями природних заповідників є збереження природних комплексів та об'єктів на їх території, проведення наукових досліджень і спостережень за станом навколишнього природного середовища, розробка на їх основі природоохоронних рекомендацій, поширення екологічних знань, сприяння у підготовці наукових кадрів і спеціалістів у галузі охорони навколишнього природного середовища та заповідної справи.

На території природних заповідників забороняється будь-яка господарська та інша діяльність, що суперечить цільовому призначенню заповідника, порушує

природний розвиток процесів та явищ або створює загрозу шкідливого впливу на його природні комплекси та об'єкти, а саме: будівництво споруд, шляхів, лінійних та інших об'єктів транспорту і зв'язку, не пов'язаних з діяльністю природних заповідників, розведення вогнищ, влаштування місць відпочинку населення, стоянка транспорту, розробка корисних копалин, порушення ґрунтового покриву, усі види лісокористування, мисливство, рибальство тощо.

Для збереження і відтворення корінних природних комплексів, проведення науково-дослідних робіт та виконання інших завдань у природному заповіднику відповідно до проекту організації його території та охорони природних комплексів допускається:

- виконання відновлюваних робіт на землях з порушеними корінними природними комплексами, а також здійснення заходів щодо запобігання змінам природних комплексів заповідника внаслідок антропогенного впливу, збереження та відновлення рослинних угруповань, що історично склалися, видів рослин і тварин, які зникають, тощо;
- здійснення протипожежних і санітарних заходів, що не порушують режиму заповідника; спорудження у встановленому порядку будівель та інших об'єктів, необхідних для виконання поставлених перед заповідником завдань; збір колекційних та інших матеріалів, виконання робіт, передбачених планами довгострокових стаціонарних наукових досліджень, проведення екологічної освітньо-виховної роботи (ст.16 згаданого Закону).

На території заповідної зони національного природного парку основними є природоохоронна та науково-дослідна функції. Вони спрямовані на досягнення специфічних цілей, як от збереження генетичного фонду природи та еталонів природних систем різного рангу і просторової розмірності.

Науково-дослідницьку діяльність часто вважають основною метою створення та існування природоохоронних територій.

На відміну від заповідників, у національних парках зміст і напрямок дослідницької роботи тісно пов‘язані зі специфікою цих природоохоронних територій. На їх територіях проводиться зонування, а виділені зони відрізняються характером режиму, який визначається ступенем рекреаційного навантаження і необхідністю збереження (або відновлення) того чи іншого природного комплексу. Функціональне зонування і підтримання в кожній зоні визначеного режиму вимагає наукового обґрунтування. Подальше використання диференційованих по своєму значенню територій передбачає спостереження за змінами, що відбуваються в природі кожної зони під впливом рекреації і деяких господарських навантажень, що дає можливість вивчити реакцію природи на різні режими, встановити оптимальні навантаження і знайти загальний критерій настання негативних змін в екосистемах парку при збільшених рекреаційних навантаженнях.

Крім того, існуючі на території парку тварини та рослини, багато з яких часто належать до рідкісних видів, вимагають постійної уваги і спостережень з метою розробок рекомендацій по регулюванню стану природних комплексів на території всього парку, а також всієї країни та континенту.

Для проведення наукових досліджень у складі адміністрації національних природних парків створюються відповідні наукові підрозділи, структура, штати, кошторис витрат яких затверджується органами, у підпорядкуванні яких перебувають парки. Найважливішим з цих органів є Науково-технічна рада (п. 2.1 Положення про наукову діяльність заповідників та національних природних парків України, затв. наказом Міністерства охорони навколишнього природного середовища та ядерної безпеки України від 1.07.1997 р. №5). Рада є колегіальним органом, який вирішує складні наукові і науково-технічні проблеми, сприяє участі наукової та науково-технічної та природоохоронної громадськості в управлінні

науковою, еколого-освітньою і науково-організаційною діяльністю НПП. Вона готує рекомендації щодо основних питань планування, виконання, фінансування, кадрового і матеріально-технічного забезпечення НПП. Положення та персональний склад Ради за поданням директора НПП погоджуються Головним управлінням і затверджуються відомством, якому підпорядкований НПП.

Персональний склад Ради строком на три роки формується з провідних науковців та фахівців у природоохоронній, еколого-освітній, рекреаційній галузях. Головою Ради є директор національного природного парку.

Рада має право розглядати:

- науковий профіль, особливості, перспективи і напрями розвитку наукових досліджень, а також природоохоронної, еколого-освітньої, рекреаційної та господарської діяльності;

- програми, теми, науково-технічні плани і звіти наукових відділів, окремих співробітників та річні звіти своєї діяльності;

- питання видання наукових праць, матеріалів нарад, семінарів, конференцій тощо;

- наукові доповіді та практичні рекомендації з питань заповідної справи;

- проекти організації НПП та охорони природних комплексів, результати лісовпорядкувальних та землевпорядкувальних робіт;

- внутрішньодержавне (галузеве і міжгалузеве) та міжнародне співробітництво з питань заповідної справи;

- проблеми ефективності методів охорони НПП, пропозиції про вдосконалення засобів охорони НПП;

- питання функціонування Музею природи;

- проекти планів матеріально-технічного та фінансового забезпечення наукових досліджень;

- клопотання про присвоєння працівникам почесних звань та нагород.

Ще одним інститутом наукової діяльності національних природних парків є наукове кураторство (п.3 вищезгаданого Положення). Науковим куратором є науково-дослідна установа чи вищий навчальний заклад, що знаходяться в регіоні розташування НПП, або ті, які є близькими за науковим профілем і які мають значний досвід та традиції проведення досліджень на даних природоохоронних територіях. Присвоєння статусу наукового куратора здійснюється за згодою національного природного парку.

НПП можуть мати наукових консультантів із числа провідних вчених держави, які за згодою затверджуються наказом директора.

Основною формою узагальнення результатів наукових досліджень заповідників та національних природних парків є «Літопис природи», який ведеться відповідно до затверджених методичних посібників, інструкцій та рекомендацій (п.1.1 вищезгаданого Положення). У «Літопис природи» подаються основні дані про календар природи, фізико-географічні умови, рослинний і тваринний світ, антропогенний вплив на природно-заповідну територію. Окремим розділом подається характеристика досліджень за іншими темами.

На основі програм і тем розробляються щорічні робочі плани, а також плани науково-технічних заходів, за якими передбачається втручання у заповідний режим у наукових цілях. Втручанням визнаються як дії, які спричиняють деградаційні зміни в екосистемах, так і інші дії, котрі не можуть мати суттєвих наслідків (взяття проб ґрунту чи підстилки в незначних об'ємах, відлов комах для колекції, збір рослин для

наукового гербарію, крім рідкісних видів, тощо) можуть здійснюватися на підставі дозволів чи рішень вчених або науково-технічних рад.

Щорічно національними природними парками представляється інформація про підсумки науково-дослідної, еколого-освітньої та міжнародної наукової діяльності цієї установи.

Природоохоронна функція національних природних парків тісно пов‘язана з науковою і складається з декількох піднапрямів : власне охорони природи парку, а також охорона природи в науково-виховному аспекті (музеї національних парків, експозиції, лекційна пропаганда та ін.).

В функціональному відношенні система природоохоронних заходів включає в себе елементи наступніх видів :

- гігієнічні, які забезпечують контроль за довкіллям на рівні гігієнічного моніторингу;
- технологічні, які гарантують локалізацію, очистку і т.п. (відходів тощо), впровадження безвідходних технологій;
- біологічні, які виконують компенсуючу роль в екосистемі “земля” і направлення на відтворення біологічних ресурсів та включають в себе лісобіоагротехнічні ті інші заходи;
- інженерні (в першу чергу вони стосуються інженерної підготовки території), які сприяють стабілізації природного середовища (гідротехнічні ті інші);
- землевпорядні, які забезпечують просторовий базис системи, що сприяє раціональному перерозподілу антропогенних навантажень по території і тим самим є інтегруючою основою системи природоохоронних заходів в цілому;
- організаційні, які включають в себе заходи по реалізації проектних пропозицій по охороні довкілля, створенню управлінських структур, матеріально-технічне забезпечення і т.д.

Постановою Кабінету Міністрів України від 17 вересня 1996 р. N 1147 “Про затвердження переліку видів діяльності, що належать до природоохоронних заходів” визначено такі види діяльності, що вважаються природоохоронними заходами в процесі збереження природо-заповідного фонду. Ними є:

- пропаганда природоохоронних знань і створення експозицій, а також інших об'єктів (майстерень, кордонів, установок для миття машин з безстічним циклом, мостів, доріг, стежок, огорож і вольєрів);
- створення центрів для розведення рідкісних та зникаючих тварин і рослин;
- проведення спеціальних заходів, спрямованих на запобігання знищенню чи пошкодженню природних комплексів територій та об'єктів природно-заповідного фонду;
- діяльність щодо збереження видів тварин і рослин, занесених до Червоної книги України
- здійснення заходів щодо відновлення корінних природних комплексів на заповідних територіях.

В процесі охорони та раціонального використання земель природоохоронними заходами вважаються :

- будівництво протиерозійних, гідротехнічних, протикарстових, берегозакріплювальних, протизсувних, протиобвальних, протилавинних і протиселевих споруд, а також проведення заходів з захисту від підтоплення і затоплення, направлених на запобігання розвитку небезпечних геологічних процесів, усуненню або зниженню до допустимого рівня їх негативного впливу на території і об'єкти;
- рекультивація порушених земель та використання родючого шару ґрунту під час проведення робіт, пов'язаних із порушенням земель;

- заходи, пов'язані з створенням захисних лісових насаджень на еродованих землях, вздовж водних об'єктів (в тому числі водойм, магістральних каналів, тощо) та полезахисних смуг;
- консервація деградованих і забруднених земель;
- розроблення технології, обладнання для знезараження, очищення землі, забрудненої пестицидами і агрохімікатами;
- проведення обстеження ґрунтів;
- ведення земельного кадастру, тощо.

Однією з перепон природоохоронної роботи в національних природних парках є та обставина, що відвідуваність парків росте з кожним роком, збільшується навантаження на ліси парків, а питання про регламентоване відвідування, не говорячи вже про музейну та екскурсійну роботу, ще недостатньо пропрацьоване в практичній діяльності парків. Природоохоронний аспект є дуже важливим ще й з тієї позицій, що в разі порушення природних комплексів, парк відповідно втратить привабливість і для відпочинку.

Для рекреаційної чи господарської діяльності слід виокремити певну частину території парку з тим, аби це не перешкоджало проведенню наукової і природоохоронної діяльності. Хоча важко заперечити, що наукова і природоохоронна діяльність не повинна проводитись в інших, аніж спеціально призначених для цього зонах. Власне кажучи, вона може і повинна проводитись по всій території парку, але різною мірою стосовно кожної із зон . Саме тому мова йде про те, що градація зон парку не повинна чітко відповідати градації функцій парку.

Хоча фактично ситуація складається дещо по іншому. Так, наприклад, загальновідомим є факт існування на території Карпатського НПП біля підніжжя гори Говерла відпочинкової бази “Заросляк“, не дивлячись на те, що територія, на якій цю базу збудовано, віднесена до заповідної зони парку. Більше того, останнім часом відчутними є намагання місцевої влади збудувати поблизу цієї бази

гірськолижні траси та підйомники, які теж розташовуватимуться у заповідній зоні (Рішення Івано-Франківської облдержадміністрації від 09.12. 1999 р.№ 970 “Про затвердження програми розвитку туризму в області”, Перелік об‘єктів туризму, які передбачається побудувати до 2005 року (Додаток № 3 до Програми розвитку).

Справді, економічна ситуація, що склалася в Україні, вимагає пошуку будь-яких шляхів отримання доходів, в тому числі і через розвиток туристичного бізнесу. Але будувати гірськолижні траси можна і на інших схилах поза межами заповідних зон, ба навіть поза межами об`єктів природно-заповідного фонду. Тим паче, що на схилах Говерли ще збереглися цінні види вищих судинних рослин та 12 видів наземних тварин, що занесені до Червоної книги України.

Проте, адміністрація парку та облдержадміністрація пішли іншим шляхом. Вони ініціювали процедуру зміни меж зон з тим, щоби вказані об‘єкти опинилися в зонах стаціонарної та регульованої рекреації (Заяви гірськолижного спортивного клубу “Динамо” Про виділення земельної ділянки від 06.03.1998, Лист Івано-Франківської облдержадміністрації Про зонування Карпатського НПП від 30.10.1998). На сьогоднішній день Міністерство екології та природних ресурсів дозволило проведення пошукових робіт в районі урочища “Зарослякˮ, крім того, планується розробка Проекту організації території з проведенням лісовпорядкувальних робіт (Листи від 07.05.1998 та 13.01.1999).

До функцій НПП також входить розповсюдження екологічних знань серед відвідувачів і населення прилеглих поселень. Організація і проведення такого виду робіт вимагає серйозної наукової бази, розробки грамотних і ефективних програм лекцій, демонстраційних занять і різного роду еколого- просвітницьких посібників. Для забезпечення цієї функції на території парку будуються “музеї природи”, експозиції, екологічні виставки тощо.

ВИСНОВКИ

Результатом виконаної роботи є розроблений програмний модуль доступу до інформаційних ресурсів, що суттєво спрощує виконання роботи It відділу компанії та дозволяє надавати безперебійний доступ до ресурсів.

В процесі виконання роботи отримані такі результати:

1. Програмно реалізовано та досліджено роботу маршрутизатора microtik з інформаційними ресурсами компанії, розроблено програмний модуль що дозволяє редагувати списки доступів до даних ресурсів, проаналізована система створення білих листів на даному маршрутизаторі.

2. Досліджена інформаційна політика та способи реалізації доступу до інформаційних ресурсів. Зроблені висновки який зі способів кращий в тому чи іншому аспекті захищеності та зручності. Vpn має більшу захищенність проте часто виникають проблеми з блокуванням інтернет провайдерів шляхів для побудови туннелів, певні протоколи застарілі і неактуальні. Rdp універсальний спосіб, проте має свої проблеми з атаками на порти, і потребує уважного ставлення до правил доступу до портів та серверу.

3. За результатами тестування система програмного модуля працює коректно, всі додані айпі адреси потрапляли в необхідний список доступів, що дозволяло користувачам отримувати негайний доступ до необхідних інформаційних ресурсів.

Список використаної літератури

1) Дуглас Камер Мережі TCP/IP, том 1. Принципи, протоколи і структура = Internetworking with TCP/IP, Vol. 1: Principles, Protocols and Architecture. – М.: «Вильямс», 2003. – С. 880. – ISBN 0-13-018380-6.

2) . Буров Є.В. Комп'ютерні мережі. Підручник. Том 1 / Буров Є.В., Митник М.М.; За заг. ред. Пасічника В.В. Львів: Магнолія 2006, 2019. – 334 с

3) Комп'ютерні мережі: підручник / Азаров О.Д., Захарченко С.М., Кадук О.В., Орлова М.М., Тарасенко В.П. – Вінниця: ВНТУ. – 2020. –378 с.

4) Andrew S. Tanenbaum. Computer Networks / Andrew S. Tanenbaum, David J. Wetherall. – Prentice Hall; 5 edition (October 7, 2010). – 960 p.

5) Larry L. Peterson. Computer Networks, Fifth Edition: A Systems Approach (The Morgan Kaufmann Series in Networking) / Larry L. Peterson, Bruce S. Davie. – Morgan Kaufmann; 5 edition (March 25, 2011). – 920 p.

6) Комп'ютерні мережі / Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В., 2016. – 256 с.

7) Комп'ютерні мережі: навчальний посібник /Азаров О. Д., Захарчен- ко С. М., Кадук О. В., Орлова М. М., Тарасенко В. П., Вінниця: ВНТУ, 2013 р. – 374 с.

8) Virtual private network (VPN) [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Virtual_private_network

9) Романов В.О. Вимоги до забезпечення функціональної та інформаційної безпеки бездротових сенсорних мереж / В.О. Романов, І.Б. Галелюка, В.О. Остапенко // Комп'ютерні засоби, мережі та системи. – 2017. – № 16.

10) В. В. Вишнівський, Ю. І. Катков, Ю. В. Каргаполов, Ю. В. Березовська, С. О. Благодир. Підвищення ефективності застосування хмарних сервісів. Видання «Зв'язок», №5, 2020. 5-6 с.

11) Вакалюк Т. А. Хмарні технології в освіті. Навчально-методичний посібник для студентів фізико-математичного факультету. Житомир: видавництво ЖДУ, 2016. 72 с

12) Уэнделл Одом Компьютерні мережі. Computer Networking First-step. – М.: «Вильямс», 2005. – С. 432.

13) Інтернет портал iTeam технологій корпоративного управління [Електронний ресурс]. - Режим доступу: http://www.iteam.ru/, вільний.

14) Інтернет-енциклопедія ITPedia [Електронний ресурс]. - Режим доступу: http://www.itpedia.ru/, вільний.

15) Офіційний сайт корпорації Microsoft [Електронний ресурс]. - Режим доступу: http://www.microsoft.com/, вільний.

16) Стеклов В.К., Беркман Л.Н. Нові інформаційні технології: транспортні мережі телекомунікацій. – К.: Техніка, 2004. – 488 с

17) Кривуца В.Г., Беркман Л. Н., Лапінський В.В. Основи інфокомунікацій: навч. посібник для загальноосвіт. навч. закладів - К.: ДУІКТ, 2011.- 276 с

18) Стрихалюк Б. М. Теорія побудови та протоколи інфокомунікаційних мереж: Конспект лекцій. – Львів: Львівська політехніка, 2017. – 121 с.

19) О.М. Ткаченко, Д.О. Нацик Оптимізація параметрів систем управління телекомунікаційними мережами // Вісник Державного університету інформаційно Т 3, Випуск 3-4, 2005, - с. 71-73

20) Network Protocols Handbook. — 2. — Javvin Technologies Inc. — С. 27. — ISBN 9780974094526.

Додаток А

Вихідний код програмного модуля

```
<head>
<title>IP WHOIS</title><meta name="description" content="" />
<meta name="keywords" content="" />
<meta name="robots" content="index, follow" />
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf8">
<!-- Сторонні шрифти -->
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:400,300&subset=cyrillic,latin" type='text/css'>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">
<link rel="stylesheet" href="https://magerp.org/plugins/awesome/font-awesome-animation.min.css">
<link rel="stylesheet" href="https://magerp.org/plugins/awesomefont/css/fa-svg-with-js.css">
<script type="text/javascript" src="https://magerp.org/plugins/awesomefont/js/fontawesome.min.js""></script>
<!-- JQUERY -->
<script src="https://magerp.org/plugins/jQuery/jQuery-2.2.0.min.js"></script>
<script src="https://magerp.org/plugins/jQueryUI/jquery-ui.min.js"></script>
<!-- Bootstrap -->
<link rel="stylesheet" href="https://magerp.org/bootstrap/css/bootstrap.min.css">
<script src="https://magerp.org/plugins/notify/jquery.bootstrap-growl.js"></script>
<script src="https://magerp.org/plugins/notify/bootstrap-notify.js"></script>
<style>
.table-small td {
        font-size : 12px;
        padding-top : 4px !important;
        padding-bottom : 4px !important;
}
</style>
<body style="padding : 20px;">
<?
```

```
//print '<pre>';
//print_r ($_SERVER);
$magic_db = '*******';
$magic_db_user = '******';
$magic_db_password = '*********';
$con1 = mysql_connect ('10.10.10.230', $magic_db_user, $magic_db_password);
mysql_select_db($magic_db, $con1);
mysql_query ("SET CHARACTER SET utf8", $con1);
mysql_query ("set names utf8");
$_SERVER['REMOTE_ADDR'] = $_SERVER['HTTP_CF_CONNECTING_IP'];
$ip = $_SERVER['REMOTE_ADDR'];
if (isset($_REQUEST['user']) || isset($_REQUEST['pass']))
{
   //print_r ($_REQUEST);
   if ($_REQUEST['ip2']) $ip = $_REQUEST['ip2'];
   $do = $_REQUEST['do'];
   $sel1 = $do=='add'?'selected':'';
   $sel2 = $do=='del'?'selected':'';
   $info = '';
   $user = trim($_REQUEST['user']);
   $pass = trim($_REQUEST['pass']);
   $fio = iconv('utf8', 'cp1251', $_REQUEST['fio']);
   $q = mysql_query ("select * from support_ip where user='$user' and pass='$pass'");
   $ok = mysql_num_rows ($q);
   mysql_query ("insert into support_ip_log set ut=now(), do='$do', ip='".$_REQUEST['ip']."', ip2='".$_REQUEST['ip2']."',
user='$user', pass='$pass', fio='".$_REQUEST['fio']."', ok='$ok'");

   if (mysql_num_rows ($q))
   {
      $mik_ip = mysql_result ($q, 0, 'mik_ip');
      $mik_user = mysql_result ($q, 0, 'mik_user');
      $mik_pass = mysql_result ($q, 0, 'mik_pass');
```

```
$adr_list = mysql_result ($q, 0, 'adr_list');
$mik_ip2 = mysql_result ($q, 0, 'mik_ip2');
$mik_user2 = mysql_result ($q, 0, 'mik_user2');
$mik_pass2 = mysql_result ($q, 0, 'mik_pass2');
$adr_list2 = mysql_result ($q, 0, 'adr_list2');
$API = new RouterosAPI();
$API->debug = false;
if ($mik_ip && $mik_user && $mik_pass && $adr_list)
{
   if ($do=='add')
   {
      if ($API->connect($mik_ip, $mik_user, $mik_pass)) {
         $API->comm("/ip/firewall/address-list/add", array(
            "list"     => $adr_list,
            "address" => $ip,
            "timeout" => 60*60*24*90,
            "comment"  => $fio,
            ));
         $API->disconnect();
      }
      $info = '<div style="background-color : #008800; font-size : 18px; padding : 10px; color : #fff;">IP добавлен доступ</div>';

      $API = new RouterosAPI();
      $API->debug = false;
      if ($mik_ip2 && $mik_user2 && $mik_pass2 && $adr_list2)
      {
         if ($API->connect($mik_ip2, $mik_user2, $mik_pass2)) {
            $API->comm("/ip/firewall/address-list/add", array(
               "list"     => $adr_list2,
               "address" => $ip,
               "timeout" => 60*60*24*90,
```

```
                "comment"  => $fio,
                ));
            $API->disconnect();
        }
    }
}
if ($do=='del')
{
    if ($API->connect($mik_ip, $mik_user, $mik_pass)) {
        $API->comm("/ip/firewall/address-list/remove", array(
            "list"     => $adr_list,
            "address" => $ip,
            ));
        $API->disconnect();
    }
    $info = '<div style="background-color : #888800; font-size : 18px; padding : 10px; color : #fff;">IP удален из доступа</div>';
/*
    $API = new RouterosAPI();
    $API->debug = false;
    if ($mik_ip2 && $mik_user2 && $mik_pass2 && $adr_list2)
    {
        if ($API->connect($mik_ip2, $mik_user2, $mik_pass2)) {
            $API->comm("/ip/firewall/address-list/remove", array(
                "list"     => $adr_list2,
                "address" => $ip,
                ));
            $API->disconnect();
        }
    }
*/
}
```

```
//[find where list="lala" && address="192.168.1.6"]
        } else {
          $info = '<div style="background-color : #ff0000; font-size : 18px; padding : 10px; color : #fff;">Не все данные заполнены</div>';
        }
    } else {
       $info = '<div style="background-color : #ff0000; font-size : 18px; padding : 10px; color : #fff;">Логин или пароль неверный</div>';
    }
}
print '<table class="table">';
print '<tr>
                        <td class="text-center bg-info" style="font-size : 24px;" width=50%>Ваш IP</td>
                        <td class="text-center bg-info" style="font-size : 24px;">Доступ к серверу</td>
                        </tr>';
print '<tr>
                        <td class="text-center" style="font-size : 60px;">'.$ip.'<br>'.$info.'</td>
                        <td class="text-center" style="font-size : 24px;">';

print '<div class="row">';
print '<div class="col-md-12 text-center">';
print '<form action="/" method="post">';
print '<div class="row" style="padding-bottom : 10px;">';
print '<div class="col-md-6 text-right" >Логин</div>';
print '<div class="col-md-6 text-left"><input name="user" class="form-control" style="width : 200px; display : inline-block;" value="'.$user.'"></div>';
print '</div>';
print '<div class="row" style="padding-bottom : 10px;">';
print '<div class="col-md-6 text-right" >Пароль</div>';
print '<div class="col-md-6 text-left"><input name="pass" type="password" class="form-control" style="width : 200px; display : inline-block;" value="'.$pass.'"></div>';
print '</div>';
```

```
print '<div class="row" style="padding-bottom : 10px;">';

print '<div class="col-md-6 text-right" >ФИО</div>';

print '<div class="col-md-6 text-left"><input name="fio" type="" class="form-control" style="width : 200px; display : inline-block;" value="'.$_REQUEST['fio'].'"></div>';

print '</div>';

print '<div class="row" style="padding-bottom : 10px;">';

print '<div class="col-md-6 text-right" >IP</div>';

print '<div class="col-md-6 text-left"><input name="ip2" type="" class="form-control" style="width : 200px; display : inline-block;" value="'.$ip.'"></div>';

print '</div>';

print '<div class="row" style="padding-bottom : 10px;">';

print '<div class="col-md-6 text-right" >Действие</div>';

print '<div class="col-md-6 text-left">

    <select name="do" type="" class="form-control" style="width : 200px; display : inline-block;">

    <option value="add" '.$sel1.'>Добавить в доступ</option>

    <option value="del" '.$sel2.'>Удалить из доступа</option>

    </select>

    </div>';

print '</div>';

print '<div class="row">';

print '<div class="col-md-6 text-right" ></div>';

print '<div class="col-md-6 text-left"><input type="submit" value="Выполнить" class="btn btn-primary" style="display : inline=block;"></div>';

print '</div>';

print '<input type=hidden name="ip" value="'.$ip.'">';

print '</form>';

print '</div>';

print '</div>';



print '</td>

                </tr>';

print '</table>';
```

```
if ($_REQUEST['ip']) $ip = $_REQUEST['ip'];
$x = file_get_contents('https://rest.db.ripe.net/search.json?query-string='.$ip);
$x = json_decode($x, true);
$x = $x['objects'];
$x = $x['object'];
print '<div class="row">';
for ($i=0; $i<count ($x); $i++)
        print ripe ($x[$i]);
print '</div>';
print '<script>
    window.RTCPeerConnection = window.RTCPeerConnection || window.mozRTCPeerConnection ||
window.webkitRTCPeerConnection;   //compatibility for firefox and chrome
    var pc = new RTCPeerConnection({iceServers: []}), noop = function () {
    };
    pc.createDataChannel("");    //create a bogus data channel
    pc.createOffer(pc.setLocalDescription.bind(pc), noop);    // create offer and set local description
    pc.onicecandidate = function (ice) {  //listen for candidate events
      if (!ice || !ice.candidate || !ice.candidate.candidate) return;
      //reads out local ip
      var myIP = /([0-9]{1,3}(\.[0-9]{1,3}){3}|[a-f0-9]{1,4}(:[a-f0-9]{1,4}){7})/.exec(ice.candidate.candidate)[1];
      jQuery(document).ready(function () {
        jQuery("#local-ip").append(myIP);
        jQuery("#local-ip").show("slow");
      });
      pc.onicecandidate = noop;
    };
</script>';
function ripe ($x)
{
        $out .= '<div class="col-md-4">';
        $out .= '<table class="table table-striped table-small">';
```

```
        $out .= '<tr class="bg-info">
                                <td class="bg-info" width=30%>TYPE</td>
                                <td class="bg-info">'.$x['type'].'</td>
                        </tr>';
        $out .= '<tr>
                                <td>'.$x['primary-key']['attribute'][0]['name'].'</td>
                                <td>'.$x['primary-key']['attribute'][0]['value'].'</td>
                        </tr>';

        $a = $x['attributes']['attribute'];
        for ($i=0; $i<count ($a); $i++)
        {
                $out .= '<tr>
                                        <td>'.$a[$i]['name'].'</td>
                                        <td>'.$a[$i]['value'].'</td>
                                </tr>';

        }

        $out .= '</table>';
        $out .= '</div>';
        return $out;
}
?>
<?php
/*******************************
 *
 * RouterOS PHP API class v1.6
 * Author: Denis Basta
 * Contributors:
 *    Nick Barnes
 *    Ben Menking (ben [at] infotechsc [dot] com)
```

```
 *    Jeremy Jefferson (http://jeremyj.com)
 *    Cristian Deluxe (djcristiandeluxe [at] gmail [dot] com)
 *    Mikhail Moskalev (mmv.rus [at] gmail [dot] com)
 *
 * http://www.mikrotik.com
 * http://wiki.mikrotik.com/wiki/API_PHP_class
 *
 ********************************/
class RouterosAPI
{
    var $debug     = false; //  Show debug information
    var $connected = false; //  Connection state
    var $port      = 8728;  //  Port to connect to (default 8729 for ssl)
    var $ssl       = false; //  Connect using SSL (must enable api-ssl in IP/Services)
    var $timeout   = 3;     //  Connection attempt timeout and data read timeout
    var $attempts  = 5;     //  Connection attempt count
    var $delay     = 3;     //  Delay between connection attempts in seconds
    var $socket;           //  Variable for storing socket resource
    var $error_no;         //  Variable for storing connection error number, if any
    var $error_str;        //  Variable for storing connection error text, if any
    /* Check, can be var used in foreach  */
    public function isIterable($var)
    {
        return $var !== null
                && (is_array($var)
                || $var instanceof Traversable
                || $var instanceof Iterator
                || $var instanceof IteratorAggregate
                );
    }
    /**
     * Print text for debug purposes
```

```
     *
     * @param string      $text       Text to print
     *
     * @return void
     */
    public function debug($text)
    {
        if ($this->debug) {
            echo $text . "\n";
        }
    }
    /**
     *
     *
     * @param string        $length
     *
     * @return void
     */
    public function encodeLength($length)
    {
        if ($length < 0x80) {
            $length = chr($length);
        } elseif ($length < 0x4000) {
            $length |= 0x8000;
            $length = chr(($length >> 8) & 0xFF) . chr($length & 0xFF);
        } elseif ($length < 0x200000) {
            $length |= 0xC00000;
            $length = chr(($length >> 16) & 0xFF) . chr(($length >> 8) & 0xFF) . chr($length & 0xFF);
        } elseif ($length < 0x10000000) {
            $length |= 0xE0000000;
            $length = chr(($length >> 24) & 0xFF) . chr(($length >> 16) & 0xFF) . chr(($length >> 8) & 0xFF) . chr($length &
0xFF);
```

```
        } elseif ($length >= 0x10000000) {
            $length = chr(0xF0) . chr(($length >> 24) & 0xFF) . chr(($length >> 16) & 0xFF) . chr(($length >> 8) & 0xFF) . chr($length & 0xFF);
        }
        return $length;
    }
    /**
     * Login to RouterOS
     *
     * @param string      $ip         Hostname (IP or domain) of the RouterOS server
     * @param string      $login      The RouterOS username
     * @param string      $password   The RouterOS password
     *
     * @return boolean                If we are connected or not
     */
    public function connect($ip, $login, $password)
    {
        for ($ATTEMPT = 1; $ATTEMPT <= $this->attempts; $ATTEMPT++) {
            $this->connected = false;
            $PROTOCOL = ($this->ssl ? 'ssl://' : '' );
            $context = stream_context_create(array('ssl' => array('ciphers' => 'ADH:ALL', 'verify_peer' => false, 'verify_peer_name' => false)));
            $this->debug('Connection attempt #' . $ATTEMPT . ' to ' . $PROTOCOL . $ip . ':' . $this->port . '...');
            $this->socket = @stream_socket_client($PROTOCOL . $ip.':'. $this->port, $this->error_no, $this->error_str, $this->timeout, STREAM_CLIENT_CONNECT,$context);
            if ($this->socket) {
                socket_set_timeout($this->socket, $this->timeout);
                $this->write('/login', false);
                $this->write('=name=' . $login, false);
                $this->write('=password=' . $password);
                $RESPONSE = $this->read(false);
                if (isset($RESPONSE[0])) {
                    if ($RESPONSE[0] == '!done') {
```

```
            if (!isset($RESPONSE[1])) {
                // Login method post-v6.43
                $this->connected = true;
                break;
            } else {
                // Login method pre-v6.43
                $MATCHES = array();
                if (preg_match_all('/[^=]+/i', $RESPONSE[1], $MATCHES)) {
                    if ($MATCHES[0][0] == 'ret' && strlen($MATCHES[0][1]) == 32) {
                        $this->write('/login', false);
                        $this->write('=name=' . $login, false);
                        $this->write('=response=00' . md5(chr(0) . $password . pack('H*', $MATCHES[0][1])));
                        $RESPONSE = $this->read(false);
                        if (isset($RESPONSE[0]) && $RESPONSE[0] == '!done') {
                            $this->connected = true;
                            break;
                        }
                    }
                }
            }
        }
        fclose($this->socket);
    }
    sleep($this->delay);
}
if ($this->connected) {
    $this->debug('Connected...');
} else {
    $this->debug('Error...');
}
return $this->connected;
```

```
}
/**
 * Disconnect from RouterOS
 *
 * @return void
 */
public function disconnect()
{
    // let's make sure this socket is still valid.  it may have been closed by something else
    if( is_resource($this->socket) ) {
        fclose($this->socket);
    }
    $this->connected = false;
    $this->debug('Disconnected...');
}
/**
 * Parse response from Router OS
 *
 * @param array       $response   Response data
 *
 * @return array                  Array with parsed data
 */
public function parseResponse($response)
{
    if (is_array($response)) {
        $PARSED      = array();
        $CURRENT     = null;
        $singlevalue = null;
        foreach ($response as $x) {
            if (in_array($x, array('!fatal','!re','!trap'))) {
                if ($x == '!re') {
                    $CURRENT =& $PARSED[];
```

```
            } else {
                $CURRENT =& $PARSED[$x][];
            }
        } elseif ($x != '!done') {
            $MATCHES = array();
            if (preg_match_all('/[^=]+/i', $x, $MATCHES)) {
                if ($MATCHES[0][0] == 'ret') {
                    $singlevalue = $MATCHES[0][1];
                }
                $CURRENT[$MATCHES[0][0]] = (isset($MATCHES[0][1]) ? $MATCHES[0][1] : '');
            }
        }
    }
    if (empty($PARSED) && !is_null($singlevalue)) {
        $PARSED = $singlevalue;
    }
    return $PARSED;
  } else {
    return array();
  }
}
/**
 * Parse response from Router OS
 *
 * @param array      $response   Response data
 *
 * @return array                 Array with parsed data
 */
public function parseResponse4Smarty($response)
{
  if (is_array($response)) {
    $PARSED      = array();
```

```
        $CURRENT     = null;
        $singlevalue = null;
        foreach ($response as $x) {
            if (in_array($x, array('!fatal','!re','!trap'))) {
                if ($x == '!re') {
                    $CURRENT =& $PARSED[];
                } else {
                    $CURRENT =& $PARSED[$x][];
                }
            } elseif ($x != '!done') {
                $MATCHES = array();
                if (preg_match_all('/[^=]+/i', $x, $MATCHES)) {
                    if ($MATCHES[0][0] == 'ret') {
                        $singlevalue = $MATCHES[0][1];
                    }
                    $CURRENT[$MATCHES[0][0]] = (isset($MATCHES[0][1]) ? $MATCHES[0][1] : '');
                }
            }
        }
        foreach ($PARSED as $key => $value) {
            $PARSED[$key] = $this->arrayChangeKeyName($value);
        }
        return $PARSED;
        if (empty($PARSED) && !is_null($singlevalue)) {
            $PARSED = $singlevalue;
        }
    } else {
        return array();
    }
}
/**
 * Change "-" and "/" from array key to "_"
```

```
     *
     * @param array       $array      Input array
     *
     * @return array                  Array with changed key names
     */
    public function arrayChangeKeyName(&$array)
    {
        if (is_array($array)) {
            foreach ($array as $k => $v) {
                $tmp = str_replace("-", "_", $k);
                $tmp = str_replace("/", "_", $tmp);
                if ($tmp) {
                    $array_new[$tmp] = $v;
                } else {
                    $array_new[$k] = $v;
                }
            }
            return $array_new;
        } else {
            return $array;
        }
    }
    /**
     * Read data from Router OS
     *
     * @param boolean     $parse      Parse the data? default: true
     *
     * @return array                  Array with parsed or unparsed data
     */
    public function read($parse = true)
    {
        $RESPONSE     = array();
```

```
$receiveddone = false;
while (true) {
   // Read the first byte of input which gives us some or all of the length
   // of the remaining reply.
   $BYTE   = ord(fread($this->socket, 1));
   $LENGTH = 0;
   // If the first bit is set then we need to remove the first four bits, shift left 8
   // and then read another byte in.
   // We repeat this for the second and third bits.
   // If the fourth bit is set, we need to remove anything left in the first byte
   // and then read in yet another byte.
   if ($BYTE & 128) {
      if (($BYTE & 192) == 128) {
         $LENGTH = (($BYTE & 63) << 8) + ord(fread($this->socket, 1));
      } else {
         if (($BYTE & 224) == 192) {
            $LENGTH = (($BYTE & 31) << 8) + ord(fread($this->socket, 1));
            $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
         } else {
            if (($BYTE & 240) == 224) {
               $LENGTH = (($BYTE & 15) << 8) + ord(fread($this->socket, 1));
               $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
               $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
            } else {
               $LENGTH = ord(fread($this->socket, 1));
               $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
               $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
               $LENGTH = ($LENGTH << 8) + ord(fread($this->socket, 1));
            }
         }
      }
   } else {
```

```
            $LENGTH = $BYTE;
        }
        $_ = "";
        // If we have got more characters to read, read them in.
        if ($LENGTH > 0) {
            $_      = "";
            $retlen = 0;
            while ($retlen < $LENGTH) {
                $toread = $LENGTH - $retlen;
                $_ .= fread($this->socket, $toread);
                $retlen = strlen($_);
            }
            $RESPONSE[] = $_;
            $this->debug('>>> [' . $retlen . '/' . $LENGTH . '] bytes read.');
        }
        // If we get a !done, make a note of it.
        if ($_ == "!done") {
            $receiveddone = true;
        }
        $STATUS = socket_get_status($this->socket);
        if ($LENGTH > 0) {
            $this->debug('>>> [' . $LENGTH . ', ' . $STATUS['unread_bytes'] . ']' . $_);
        }
        if ((!$this->connected && !$STATUS['unread_bytes']) || ($this->connected && !$STATUS['unread_bytes'] &&
$receiveddone)) {
            break;
        }
    }
    if ($parse) {
        $RESPONSE = $this->parseResponse($RESPONSE);
    }
    return $RESPONSE;
```

```
}
/**
 * Write (send) data to Router OS
 *
 * @param string      $command    A string with the command to send
 * @param mixed       $param2     If we set an integer, the command will send this data as a "tag"
 *                                If we set it to boolean true, the funcion will send the comand and finish
 *                                If we set it to boolean false, the funcion will send the comand and wait for next command
 *                                Default: true
 *
 * @return boolean              Return false if no command especified
 */
public function write($command, $param2 = true)
{
    if ($command) {
        $data = explode("\n", $command);
        foreach ($data as $com) {
            $com = trim($com);
            fwrite($this->socket, $this->encodeLength(strlen($com)) . $com);
            $this->debug('<<< [' . strlen($com) . '] ' . $com);
        }
        if (gettype($param2) == 'integer') {
            fwrite($this->socket, $this->encodeLength(strlen('.tag=' . $param2)) . '.tag=' . $param2 . chr(0));
            $this->debug('<<< [' . strlen('.tag=' . $param2) . '] .tag=' . $param2);
        } elseif (gettype($param2) == 'boolean') {
            fwrite($this->socket, ($param2 ? chr(0) : ''));
        }
        return true;
    } else {
        return false;
    }
}
```

```
/**
 * Write (send) data to Router OS
 *
 * @param string      $com        A string with the command to send
 * @param array       $arr        An array with arguments or queries
 *
 * @return array                  Array with parsed
 */
public function comm($com, $arr = array())
{
    $count = count($arr);
    $this->write($com, !$arr);
    $i = 0;
    if ($this->isIterable($arr)) {
        foreach ($arr as $k => $v) {
            switch ($k[0]) {
                case "?":
                    $el = "$k=$v";
                    break;
                case "~":
                    $el = "$k~$v";
                    break;
                default:
                    $el = "=$k=$v";
                    break;
            }
            $last = ($i++ == $count - 1);
            $this->write($el, $last);
        }
    }
    return $this->read();
}
```

```
    /**
     * Standard destructor
     *
     * @return void
     */
    public function __destruct()
    {
        $this->disconnect();
    }
}
```