Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання
(повна назва)
Кафедра *Інформаційно-мережної інженерії*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Реалізація захисту корпоративної інформації за допомогою протоколів VPN
(тема)

Виконав:
здобувач *4* року навчання,
групи *ТРІМІз-22-1*
Кирило СІМОНОВ
(власне ім'я, прізвище)

Спеціальність *172 Телекомунікації та радіотехніка*
(код і повна назва спеціальності)
Тип програми *освітньо-професійна*
(освітньо-професійна або освітньо-наукова)
Освітня програма *Інформаційно-мережна інженерія*
(повна назва освітньої програми)
Керівник *доц. Наталія ХАРЧЕНКО*
(посада, власне ім'я, прізвище)

Допускається до захисту
В.о. завідувача кафедри _____ *Дарія ЧЕБОТАРЬОВА*
(підпис) (власне ім'я, прізвище)

2026 р.

Не містить відомостей заборонених до відкритого публікування.

Здобувач / Кирило Сімонов /

Керівник / Наталія Харченко /

Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання
Кафедра *Інформаційно-мережної інженерії*
Рівень вищої освіти *перший (бакалаврський)*
Спеціальність *172 Телекомунікації та радіотехніка*
(код і повна назва)
Тип програми *освітньо-професійна*
Освітня програма *«Інформаційно-мережна інженерія»*
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві *Сімонову Кирилу Сергійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Реалізація захисту корпоративної інформації за допомогою протоколів VPN*
затверджена наказом університету від 18 травня 2026 р. № 73 Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 липня 2026 р.

3. Вихідні дані до роботи *Провести аналіз основних характеристик корпоративних комп'ютерних мереж, визначити основні вимоги до них. Дослідити принцип роботи та особливості технології віртуальних приватних мереж. Проаналізувати протоколи VPN та визначити найбільш оптимальний для використання у корпоративній мережі. На базі проведеного аналізу впровадити обраний протокол у мережі підприємства, що складається з центрального офісу і двох віддалених магазинів, підключених через мережі провайдера.*
.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1. Основні відомості про корпоративні мережі та постановка завдання
2. Застосування технології VPN у корпоративній мережі підприємства
3. Формування системи моніторингу корпоративної мережі
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) *назва, мета і актуальність кваліфікаційної роботи; проектована корпоративна мережа; способи організації мережі VPN; порівняльний аналіз протоколів VPN; впровадження OpenVPN; створення зашифрованих тунелів SSH; оцінка продуктивності: час затримки, пропускна здатність каналу, завантаження ЦП; комплексна система моніторингу; фільтрація та аналіз веб-активності; висновки*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Ознайомлення із завданням. Уточнення ТЗ.*	*18.05.26*	*виконано*
2	*Підбір літератури за темою роботи.*	*19.05-15.06.26*	*виконано*
3	*Основні відомості про корпоративні мережі та постановка завдання*	*16.06-26.06.26*	*виконано*
4	*Застосування технології VPN у корпоративній мережі підприємства*	*27.06-03.07.26*	*виконано*
5	*Формування системи моніторингу корпоративної мережі*	*04.07-10.07.26*	*виконано*
6	*Оформлення презентаційного матеріалу, підготовка до захисту в ЕК*	*11.07-14.07.26*	*виконано*

Дата видачі завдання _18_ ___травня__ 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Наталія Харченко
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка 92 с., 22 рис., 3 табл., 10 джерел, 4 додатки.

Об'єкт дослідження – корпоративна мережа.

Мета роботи – забезпечення захисту інформації за допомогою протоколів VPN.

Розглянуто особливості побудови сучасних корпоративних мереж, їх характеристики та основні вимоги. Проаналізовано варіанти передачі інформації між віддаленими офісами однієї мережі та методи її захисту на базі технології VPN. Досліджено основні протоколи VPN, визначено їх особливості та проведено порівняльний аналіз. Проведено впровадження та настроювання протоколів OpenVPN та SSH у корпоративних мережах.

КОРПОРАТИВНА МЕРЕЖА, ЗАХИСТ ІНФОРМАЦІЇ, VPN, OPEN VPN, SSH, PPTP, IPSEC, L2TP.

THE ABSTRACT

Explanatory slip 92 p., 22 fig., 3 tab., 10 sources, 4 attach.

Object of research - the corporate network.

The purpose of the work - providing information security through VPN protocols.

Features of construction of modern corporate networks, their characteristics and basic requirements are considered. Options for information transfer between remote offices of one network and methods of its protection based on VPN technology are analyzed. The basic VPN protocols are investigated, their features are defined and the comparative analysis is carried out. Implementation and configuration of OpenVPN and SSH protocols in corporate networks are carried out.

CORPORATE NETWORK, PROTECTION OF INFORMATION, VPN, OPEN VPN, SSH, PPTP, IPSEC, L2TP.

ЗМІСТ

С.

ПЕРЕЛІК СКОРОЧЕНЬ 9
ВСТУП 11
1 ОСНОВНІ ВІДОМОСТІ ПРО КОРПОРАТИВНІ МЕРЕЖІ ТА ПОСТАНОВКА ЗАВДАННЯ 13
1.1 Основні відомості про корпоративні мережі 13
1.1.1 Організація зв'язку 13
1.1.2 Визначення корпоративної мережі 15
1.1.3 Структура корпоративної мережі 16
1.2 Основні відомості про віртуальні приватні мережі 19
1.2.1 Організація VPN 19
1.2.2 Протоколи VPN. OpenVPN 22
1.2.3 Протокол SSH 25
1.3 Система моніторингу корпоративних мереж 26
1.4 Особливості та функції корпоративної мережі 27
2 ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ VPN У КОРПОРАТИВНІЙ МЕРЕЖІ ПІДПРИЄМСТВА 29
2.1 Впровадження технології OpenVPN 29
2.2 Реалізація на основі технології SSH 40
2.2.1 Формування SSH-тунелю 40
2.2.2 Створення захищеного тунелю SSH VPN 42
2.3 Оцінка продуктивності сформованої корпоративної мережі 43
2.3.1 Оцінка продуктивності мережі з технологією OpenVPN 44
2.3.2 Оцінка продуктивності мережі з технологією SSH 46
2.3.3 Порівняння SSH та OpenVPN 48
3 ФОРМУВАННЯ СИСТЕМИ МОНІТОРИНГУ КОРПОРАТИВНОЇ МЕРЕЖІ 50
3.1 Система моніторингу серверної інфраструктури Nagios 50
3.2 Система моніторингу продуктивності серверів Cacti 58
3.3 Методи аналізу трафіку в корпоративній мережі 63
3.3.1 Використання Proxy-серверу 63

3.3.2 Настроювання аналізатора логів проксі-сервера 67
3.4 Аналіз трафіку корпоративної мережі 70
ВИСНОВКИ 75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 76
ДОДАТОК А ЗНАЧЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ ІНТЕРНЕТ-КАНАЛУ ПРИ ВИКОРИСТАННІ OPEN VPN 77
ДОДАТОК Б ЗНАЧЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ ІНТЕРНЕТ-КАНАЛУ ПРИ ВИКОРИСТАННІ SSH 79
ДОДАТОК В КОД НАСТРОЮВАННЯ ВЕБ-ІНТЕРФЕЙСУ 81
ДОДАТОК Г СЛАЙДИ ПРЕЗЕНТАЦІЇ 85

ПЕРЕЛІК СКОРОЧЕНЬ

AH – (Authentication Header) заголовок аутентифікації;

CentOS – (Community ENTerprise Operating System) дистрибутив Linux, заснований на комерційному Red Hat Enterprise Linux;

CPE – (Customer Premises Equipment) абонентське телекомунікаційне обладнання;

DCE – (Data Circuit terminating Equipment) пристрої, які забезпечують необхідний протокол фізичного рівня;

DTE – (Data Terminal Equipment) пристрої вироблення даних для передачі в ТМЗ;

ESP – (Encapsulating Security Payload) інкапсуляція зашифрованих даних;

HTML – (Hypertext Markup Language) стандартна мова розмітки для створення веб-сторінок;

HTTP – (HyperText Transfer Protocol) протокол прикладного рівня передачі даних;

IKE – (Internet Key Exchange) схема обміну ключами через Інтернет

LAN – (Local Area Network) – локальна обчислювальна мережа;

NAT – (Network Address Translation – перетворення (трансляція) мережних адрес) механізм зміни мережної адреси в заголовках IP датаграм;

PKI – (Public key infrastructure) інфраструктури відкритих ключів;

PPTP – (Point-to-Point Tunneling Protocol) двохточковий тунельний протокол;

POP – (Post Office Protocol – Поштовий Офісний Протокол) протокол, що використовується клієнтом для доступу до повідомлень електронної пошти;

SA – (security association) профіль безпеки;

SMTP – (Simple Mail Transfer Protocol) простий протокол пересилання пошти) протокол, який використовується для пересилання електронної пошти;

SNMP – (Simple Network Management Protocol – простий протокол керування мережею) протокол керування мережами зв'язку на основі архітектури TCP/IP;

SSTP – (Secure Socket Tunneling Protocol) протокол тунелювання розроблений Microsoft;

SSH – (Secure SHell - «безпечна оболонка») мережний протокол сеансового рівня, що дозволяє робити віддалене керування операційною системою й тунелювання;

SSL – (англ. Secure Sockets Layer – рівень захищених сокетів) криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером;

TAP або TUN – віртуальні мережні драйвери ядра системи;

TCP – (Transmission Control Protocol) протокол передачі даних із установленням з'єднання;

TLS – (Transport Layer Security – захист на транспортному рівні) криптографічний протокол, що надає можливості безпечної передачі даних в Інтернет;

UDP – (User Datagram Protocol) протоколо користувальницьких датаграм;

VPN – (Virtual Private Network) віртуальна приватна мережа;

WAN – (Wide Area Network) глобальна комп’ютерна мережа.

ВСТУП

Сучасні бізнес-процеси дедалі частіше базуються на електронному документообігу та обміні корпоративними даними. Для забезпечення цих завдань застосовується широкий спектр протоколів та технологічних рішень: від використання спеціалізованих платформ до стандартів передачі даних електронною поштою (POP, IMAP, SMTP), протоколів передачі файлів (FTP) та веб-технологій.

Інформаційні технології XXI століття сприяють оптимізації діяльності підприємств, забезпечуючи автоматизацію процесів, підвищення продуктивності праці та зниження операційних витрат. Завдяки розвитку мережевих технологій стало можливим ефективне функціонування територіально розподілених структур. Проте інтеграція таких систем потребує зваженого підходу до питань масштабованості, безпеки, продуктивності та надійності.

Ключовим викликом сучасності є захист корпоративної інформації. Передача даних через відкриті мережі несе ризики перехоплення (зокрема, з використанням аналізаторів трафіку) та несанкціонованого доступу до конфіденційних документів чи облікових даних користувачів. Традиційні методи захисту, такі як використання паролів або базове шифрування файлів, мають обмежену ефективність: вони або вразливі до криптоаналізу, або негативно впливають на швидкодію процесів через високу обчислювальну складність [1].

Оптимальним рішенням для забезпечення безпеки корпоративних мереж є впровадження технології Virtual Private Network (VPN). Використання VPN дозволяє створювати захищені віртуальні канали поверх мережі Інтернет, об’єднуючи віддалені сегменти локальних мереж. Перевагою цього підходу є шифрування трафіку на канальному рівні моделі OSI та інкапсуляція даних, що унеможливлює ідентифікацію кінцевого адресата зловмисниками [1].

Поряд із забезпеченням безпеки, критично важливою є підтримка безперебійної роботи IT-інфраструктури. Моніторинг стану серверного обладнання та сервісів у режимі реального часу дозволяє мінімізувати час простою мережі. Впровадження автоматизованих систем спостереження, що

забезпечують оперативне інформування адміністраторів, є необхідною умовою ефективного управління IT-середовищем.

З огляду на зазначене, проектування корпоративних мереж потребує комплексного аналізу. У межах даної кваліфікаційної роботи розглядаються завдання вибору та реалізації протоколів VPN, оцінка ефективності захищених каналів передачі даних, а також впровадження засобів моніторингу для забезпечення стабільної роботи мережевої інфраструктури підприємства.

1 ОСНОВНІ ВІДОМОСТІ ПРО КОРПОРАТИВНІ МЕРЕЖІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні відомості про корпоративні мережі

1.1.1 Організація зв'язку

Перш ніж аналізувати концепцію корпоративної мережі, необхідно сформулювати її базове визначення. З огляду на надмірне використання цього терміна, що призвело до певної розмитості його змісту, у межах цього контексту ми трактуємо корпоративну мережу як систему, що забезпечує інформаційну взаємодію між прикладними програмними засобами корпоративного рівня. Спираючись на це визначення, доцільно розглянути підходи до проектування таких систем, орієнтуючись на максимальну універсальність та забезпечення ефективної інтеграції поточних і перспективних додатків із мінімізацією витрат та технічних обмежень.

Корпоративна мережа, як правило, характеризується територіальною розподіленістю, об'єднуючи підрозділи та офіси, що функціонують у різних географічних локаціях. Принципи архітектури таких систем суттєво відрізняються від локальних мереж, передусім через використання орендованих каналів зв’язку з обмеженою пропускною здатністю. Якщо в межах локальних мереж основними статтями витрат є капітальні інвестиції в обладнання та інфраструктуру, то в територіально розподілених мережах ключовим фактором вартості стає регулярна плата за оренду каналів, яка пропорційна параметрам швидкості та якості передачі даних. Ця особливість є критичною і вимагає впровадження технічних рішень для мінімізації обсягів трафіку. Водночас мережева архітектура має залишатися нейтральною щодо методів обробки інформації прикладним програмним забезпеченням [1].

У даному контексті під додатками слід розуміти як системне програмне забезпечення (СУБД, поштові сервери, обчислювальні потужності, файлові системи тощо), так і прикладне ПЗ для кінцевих користувачів. Відповідно, ключовими завданнями корпоративної мережі є забезпечення інтеграції системних сервісів між територіально віддаленими вузлами та надання віддаленого доступу до них користувачам.

Побудова корпоративної мережі розпочинається з організації каналів зв’язку. У межах одного міста доцільно орендувати виділені високошвидкісні лінії, проте при об’єднанні географічно віддалених підрозділів вартість таких послуг суттєво зростає, при цьому показники якості та надійності часто залишаються незадовільними. Оптимальним рішенням у таких випадках є використання існуючої глобальної мережевої інфраструктури. Це дозволяє забезпечити комунікацію між офісами та найближчими мережевими вузлами, делегуючи завдання передачі даних між ними операторам глобальної мережі. Навіть при створенні локальних рішень варто зважати на можливість подальшого масштабування та обирати технології, сумісні з глобальними мережевими стандартами [1].

Часто як основу для корпоративної мережі розглядається інфраструктура Інтернету. Однак важливо розуміти, що Інтернет є не єдиною мережею, а «міжмережею» (internetwork). Функціонування цього середовища базується на взаємодії численних незалежних вузлів, об’єднаних через різнорідні канали передачі даних, що позбавляє структуру централізованого контролю.

Питання безпеки залишається критичним аспектом при використанні Інтернету для корпоративних потреб. Непередбачуваність маршрутизації даних крізь незалежні вузли створює ризики несанкціонованого доступу та перехоплення інформації, причому визначення джерела витоку стає технічно складним завданням. Хоча застосування засобів шифрування частково нівелює ці загрози, децентралізована природа Інтернету ускладнює адміністрування доступу. Попри використання міжмережевих екранів (firewalls), вони не можуть вважатися абсолютною гарантією захисту. Крім того, існує вразливість перед зовнішнім втручанням у роботу вузлів або маніпуляціями з архітектурними особливостями, що призводить до порушення доступності сервісів [2].

З огляду на вищезазначене, використання Інтернету як базової платформи для систем, що потребують високого рівня надійності та конфіденційності, є недоцільним. Підключення корпоративної мережі до Інтернету є обґрунтованим лише у разі потреби в забезпеченні доступу до глобальних інформаційних ресурсів.

1.1.2 Визначення корпоративної мережі

Корпоративна мережа являє собою комплексну інфраструктуру, до складу якої входять різнорідні компоненти: обчислювальна техніка (від персональних комп'ютерів до серверного обладнання), системне та прикладне програмне забезпечення, мережеві адаптери, комутаційне та маршрутизаційне обладнання, а також кабельні системи [1].

Ефективність функціонування сучасних промислових, фінансових та інших організацій значною мірою залежить від наявності цілісного інформаційного простору. Розвинена інформаційна система забезпечує оперативну обробку значних потоків даних між співробітниками, що дозволяє приймати виважені управлінські рішення для забезпечення конкурентоспроможності підприємства.

Корпоративна мережа є складною багаторівневою системою, призначеною для передачі різнотипних даних між додатками, що функціонують у межах єдиної інформаційної екосистеми організації.

Впровадження корпоративної мережі дозволяє централізувати базу даних для всіх структурних підрозділів, автоматизувати електронний документообіг, забезпечити проведення селекторних нарад та відеоконференцій з віддаленими офісами. Крім того, така мережа задовольняє потреби організації у якісному телефонному та факсимільному зв'язку (місцевому, міжміському та міжнародному), а також забезпечує захищений доступ до мережі Інтернет та інших інтерактивних ресурсів. Завдяки цьому скорочується час реагування на зміни бізнес-середовища та оптимізується управління процесами в режимі реального часу. Додатковою перевагою є зниження залежності від зовнішніх операторів фіксованого та мобільного зв'язку, що призводить до суттєвої оптимізації операційних витрат. Використання корпоративної мережі гарантує високий рівень безпеки при передачі конфіденційної виробничої та фінансової інформації, обмежуючи доступ до неї виключно авторизованими користувачами. Узагальнену схему корпоративної мережі наведено на рис. 1.1.

Основним завданням системних інтеграторів та адміністраторів є забезпечення ефективного функціонування комплексних і капіталомістких систем для обробки інформаційних потоків між співробітниками підприємства. Це сприяє прийняттю своєчасних та обґрунтованих управлінських рішень, що є критично важливим для підтримки конкурентоспроможності бізнесу. З огляду

на динамічність ринкового середовища, зміст корпоративної інформації, інтенсивність її потоків та методи обробки постійно трансформуються. Використання технологій Інтернет, як доступного та економічно доцільного інструменту, значно спростило побудову територіально розподілених корпоративних мереж. Водночас це актуалізувало питання забезпечення інформаційної безпеки під час передачі даних через загальнодоступні мережі [2].



Рисунок 1.1 - Загальна схема корпоративної мережі

1.1.3 Структура корпоративної мережі

На рис. 1.2 наведено приклад можливої архітектури корпоративної мережі.



Рисунок 1.2 - Приклад структури корпоративної мережі

Нижче наведено перелік ключових технічних засобів та інфраструктурних елементів мережі [2]:

1. Робоча станція (ПК): стаціонарний обчислювальний пристрій, підключений до локальної мережі за допомогою кабельного з’єднання (вита пара). Робоча станція оснащена необхідним програмним забезпеченням для виконання посадових обов'язків (офісні пакети, системи обліку, поштові клієнти) та інструментами для віддаленого адміністрування.

2. Мобільний пристрій: категорія техніки, що включає ноутбуки, планшети та смартфони. Основною перевагою є мобільність за умови підключення через бездротові мережі. Функціональні можливості ідентичні стаціонарним ПК.

3. Багатофункціональний пристрій (БФП): мережеве периферійне обладнання, що об'єднує функції принтера, сканера та копіювального апарата. Підключення здійснюється безпосередньо до мережі через мережевий інтерфейс або локально через хост-комп'ютер із подальшим наданням спільного доступу на рівні операційної системи.

4. Точка доступу Wi-Fi: мережеве обладнання для розгортання бездротової інфраструктури, що забезпечує підключення портативних пристроїв. Перевагою є відсутність кабельних комунікацій та мобільність користувачів. Пропускна здатність залежить від актуального стандарту передачі даних, а рівень безпеки забезпечується протоколами шифрування WPA, WPA2 та аналогами.

5. Система IP-телефонії: технологія передачі голосового сигналу через мережі IP. Оцифрування та стиснення голосових даних дозволяють оптимізувати трафік та ефективно використовувати канали передачі даних.

6. Комутатор (Switch): пристрій для об'єднання вузлів мережі в межах одного або кількох сегментів. Забезпечує адресну доставку даних безпосередньому отримувачу (окрім широкомовного трафіку), що суттєво підвищує продуктивність мережі та рівень її безпеки шляхом ізоляції нерелевантного трафіку.

7. Маршрутизатор (Router): пристрій, що виконує передачу пакетів між різними сегментами мережі на основі інформації про її топологію. Слугує шлюзом між локальною мережею та мережею Інтернет. Класифікується на

апаратні (hardware) рішення з вбудованим ПЗ та програмні (software) рішення на базі серверів (Linux/Unix) з налаштованими правилами Firewall. Вибір архітектури залежить від масштабу мережі, вимог до продуктивності та надійності системи.

8. Термінальний сервер. Високопродуктивний сервер (або кластер), що надає клієнтським станціям обчислювальні ресурси (процесорні потужності, оперативну пам'ять, дисковий простір) для виконання прикладних завдань. Архітектура передбачає підключення термінальних клієнтів, як правило малопотужних або застарілих робочих станцій, до сервера, що забезпечує віддалений доступ до робочого середовища користувача.

Переваги:

- оптимізація витрат часу на адміністрування;

- підвищення рівня інформаційної безпеки та зниження ризиків несанкціонованого доступу;

- економічна ефективність за рахунок централізації апаратних та програмних ресурсів.

Найчастіше для реалізації термінальних рішень використовується ОС Microsoft Windows Server.

9. Файловий сервер. Спеціалізований сервер, оптимізований для інтенсивних операцій введення-виведення, призначений для централізованого зберігання файлових даних. Завдяки наявності значного обсягу дискового простору та вбудованих RAID-контролерів забезпечується висока швидкість читання/запису та відмовостійкість системи.

10. Web-сервер. Програмно-апаратний комплекс, що обробляє HTTP-запити від клієнтських додатків (веб-браузерів) та надає відповідну інформацію (HTML-сторінки, медіа-контент, файли тощо). Застосовується для хостингу корпоративних ресурсів та веб-сервісів. Як правило, функціонує на базі ОС сімейства Linux/Unix, що характеризуються високою надійністю та продуктивністю, з використанням спеціалізованого ПЗ, такого як Apache або nginx.

11. Сервер додатків. Програмна платформа, призначена для забезпечення виконання бізнес-логіки додатків та надання розробникам доступу до системних компонентів через API.

12. Сервер баз даних (СУБД). Сервер, призначений для управління

базами даних, забезпечення цілісності та безпеки інформації, а також опрацювання запитів до неї. Для взаємодії з даними більшість систем використовують мову SQL (Structured Query Language).

Ключові функції SQL:

- проєктування структури баз даних та таблиць;
- виконання операцій маніпулювання даними (вставка, модифікація, видалення);
- виконання складних аналітичних запитів.

Особливістю SQL є декларативний підхід: користувач описує необхідний результат, тоді як СУБД самостійно визначає найбільш ефективні методи виконання запиту.

1.2 Основні відомості про віртуальні приватні мережі

1.2.1 Організація VPN

В умовах інтенсивного розвитку інформаційних технологій доцільність впровадження віртуальних приватних мереж (VPN) є незаперечною. Перш ніж проаналізувати ключові методи організації таких мереж, доцільно визначити їх сутність.

Віртуальна приватна мережа (Virtual Private Network, VPN) - це технологія, що забезпечує обмін даними з віддаленою локальною мережею через віртуальний канал у межах мережі загального користування з імітацією прямого з'єднання («точка-точка»). Під мережею загального користування в цьому контексті розуміють як мережу Інтернет, так і інші інтрамережі [3].

Технологія VPN надає віддаленим користувачам після проходження процедури автентифікації можливість повноцінної роботи з корпоративними ресурсами на рівні з клієнтами, що перебувають у межах центрального офісу. Центральна мережа організації забезпечує автентифікацію користувачів, навіть якщо доступ здійснюється через публічні канали зв'язку [3].

Ефективно спроектована VPN забезпечує низку стратегічних переваг, зокрема [4]:

- розширення географії доступу співробітників до корпоративної інфраструктури;
- підвищення рівня інформаційної безпеки при передачі даних;

- оптимізацію експлуатаційних витрат порівняно з традиційними глобальними мережами;

- підвищення продуктивності праці;

- спрощення топології мережі;

- підвищення мобільності персоналу та впровадження гнучких графіків роботи.

До ключових характеристик VPN належать [4]:

- безпека;

- надійність;

- масштабованість;

- керованість.

Далі розглянемо варіант організації мережевої інфраструктури установи з філіями за допомогою технології VPN (рис. 1.3).

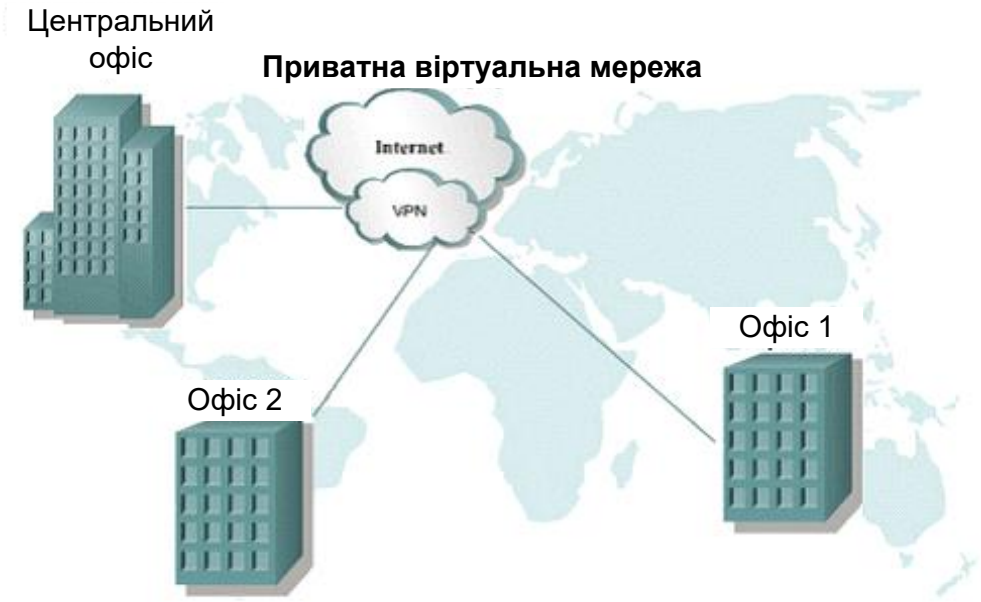


Рисунок 1.3 - Організація мережі установи з філіями з використанням VPN

Ключові особливості організації VPN-мережі [4]:

- швидкість передачі даних: попри те, що локальні мережі (100 Мбіт/с) демонструють вищі показники швидкості, можливостей сучасних інтернет-провайдерів цілком достатньо для ефективного виконання стандартних корпоративних завдань, таких як доступ до внутрішніх ресурсів або обмін

електронною кореспонденцією;

- інформаційна безпека: використання публічних каналів зв'язку потребує впровадження надійних механізмів захисту. Сучасні протоколи шифрування забезпечують високий рівень стійкості до кібератак, гарантуючи конфіденційність даних;

- економічна ефективність: завдяки гнучким тарифним планам інтернет-провайдерів витрати на обслуговування мережевої інфраструктури є оптимізованими та доступними;

- масштабованість: архітектура системи дозволяє легко інтегрувати нові філії або надавати віддалений доступ новим співробітникам без необхідності суттєвих інвестицій у додаткову комунікаційну інфраструктуру;

- гнучкість доступу: технологія VPN забезпечує безперебійний доступ до корпоративних ресурсів незалежно від географічного розташування користувача, що сприяє ефективній віддаленій роботі.

Способи реалізації VPN [5]:

1. Програмно-апаратне забезпечення: використання спеціалізованих комплексів, що забезпечують максимальну продуктивність та високий рівень захисту мережі.

2. Програмне рішення: впровадження спеціалізованого програмного забезпечення на базі існуючих робочих станцій для забезпечення функціональності VPN.

3. Інтегровані рішення: використання комплексних систем, які одночасно забезпечують роботу VPN, фільтрацію трафіку, функціонування міжмережевого екрана (firewall) та контроль якості обслуговування (QoS).

Організацію віртуальних приватних мереж (VPN) доцільно класифікувати за чотирма основними моделями:

1. Віддалений доступ (Remote Access VPN)

Модель передбачає підключення окремих співробітників до корпоративної мережі через загальнодоступні канали зв'язку. Архітектура базується на розміщенні VPN-сервера в центральному офісі, до якого ініціюють підключення віддалені клієнти. Такий підхід забезпечує мобільність персоналу, дозволяючи працювати з корпоративними ресурсами з будь-якої точки світу, де наявний доступ до мережі Інтернет.

2. Об'єднання територіально розподілених мереж (Intranet VPN)

Цей метод застосовується для створення єдиного інформаційного простору між філіями компанії. Для реалізації моделі необхідне розгортання VPN-шлюзів у кожній із мереж, що об’єднуються. Таке рішення забезпечує надійну інтеграцію філій та мобільних офісів із центральним вузлом та гарантує ефективний обмін даними між усіма сегментами організації.

3. Взаємодія із зовнішніми контрагентами (Extranet VPN)

Модель призначена для надання безпечного доступу клієнтам або партнерам компанії до специфічних корпоративних ресурсів. У контексті розвитку електронної комерції Extranet VPN забезпечує конфіденційність передачі даних, обмежуючи доступ зовнішніх користувачів виключно необхідними сервісами (наприклад, порталами з комерційними пропозиціями). Захист інформації в даному випадку реалізується через використання протоколів шифрування.

4. Вузлове шифрування (Client/Server VPN)

Метод забезпечує захист даних під час обміну між двома окремими вузлами всередині корпоративної мережі. Дана технологія використовується для сегментації трафіку в межах одного фізичного сегмента мережі, забезпечуючи логічне відокремлення даних, наприклад, між різними підрозділами організації (фінансовим департаментом та відділом кадрів). На відміну від технології VLAN, де розділення базується на налаштуваннях комутації, Client/Server VPN досягає аналогічної ізоляції завдяки криптографічному захисту трафіку.

1.2.2 Протоколи VPN. OpenVPN

Оскільки передача даних у віртуальних приватних мережах (VPN) відбувається через публічну інфраструктуру, забезпечення їхньої конфіденційності та цілісності є критично важливим завданням.

Захист інформації в межах VPN реалізується шляхом використання комбінації протоколів двох типів:

- протоколи тунелювання, що забезпечують формування VPN-з'єднання;
- протоколи шифрування, що гарантують захист даних усередині створеного тунелю.

Для порівняння ефективності реалізації VPN доцільно розглянути стандарт IPsec та технологію OpenVPN [5].

Традиційно для об'єднання територіально розподілених мереж застосовується стандарт IPsec. Проте він має низку суттєвих недоліків: складність архітектури, труднощі при налаштуванні, неоднорідність реалізацій залежно від виробника обладнання, а також потенційні вразливості та конфлікти з налаштуваннями міжмережевих екранів (firewall). Ці обставини стимулювали розвиток сучасних VPN-технологій, зокрема проекту з відкритим вихідним кодом OpenVPN, функціональні можливості якого у версії 2.0 значно перевищують класичні рішення.

OpenVPN є перспективною альтернативою IPsec. Для організацій, де використання IPsec не є обов'язковою нормативною вимогою, впровадження OpenVPN дозволяє отримати доступ до ширшого спектра можливостей за менших витрат. Серед ключових переваг: захищений віддалений доступ до корпоративних ресурсів, ефективне тунелювання через firewall та проксі-сервери, а також передача широкомовного трафіку. Використання перевірених стандартів (SSL/TLS, пристрої Tap/Tun) робить OpenVPN однією з найбільш надійних та доступних технологій на сучасному ринку [5].

OpenVPN - це інструмент з відкритим кодом, створений у 2002 році, призначений для побудови VPN-мереж типу site-to-site з використанням протоколів SSL/TLS або статичних ключів. Технологія забезпечує створення безпечного тунелю для передачі даних через TCP або UDP порти в середовищі загальнодоступних мереж [5].

Ключові технічні характеристики OpenVPN [5]:

- простота розгортання: інструмент вирізняється високим рівнем зручності під час інсталяції та конфігурації;

- кросплатформність: підтримується більшістю сучасних операційних систем, включаючи Linux (ядро 2.4+), Windows (2000/XP/Vista та новіші), OpenBSD, FreeBSD, NetBSD, Mac OS X та Solaris. Принципи налаштування є уніфікованими для всіх платформ;

- архітектура: реалізована за моделлю «клієнт-сервер». ПЗ встановлюється на всіх вузлах мережі, де один вузол виконує роль сервера, а інші – клієнтів;

- гнучкість налаштувань: стандартним портом для роботи OpenVPN є UDP 1194, проте конфігурація дозволяє використовувати будь-який інший порт. Починаючи з версії 2.0, один порт може обслуговувати кілька тунелів

одночасно.

Налаштування VPN-мережі типу Ethernet (міст) або IP (маршрутизація) реалізується за допомогою мережевих драйверів TAP або TUN. Дані інтерфейси є кросплатформними та інтегровані в ядро Linux, починаючи з версії 2.4. Детальна технічна документація щодо конфігурації доступна у відповідних сторінках довідкового посібника (man pages).

Використання статичних ключів для шифрування та дешифрування даних спрощує процес налаштування VPN-шлюзів, проте створює критичні вразливості в управлінні ключами та їх безпеці. Компрометація ключа безпосередньо призводить до розкриття захищених даних.

Для мінімізації цих ризиків рекомендується впровадження інфраструктури відкритих ключів (PKI). У цій моделі кожен вузол використовує пару ключів: публічний, доступний для всіх учасників, та приватний, що зберігається в секреті. Дана архітектура, реалізована через інтегровану в OpenVPN бібліотеку OpenSSL, забезпечує надійну автентифікацію вузлів перед ініціацією передачі даних. Використання PKI є стандартом безпеки для сучасних VPN-рішень [6].

Історично протокол IPSec був єдиним інструментом для забезпечення безпеки мереж типу site-to-site та клієнт/сервер. З розвитком протоколу SSL ситуація змінилася: розроблений спочатку для захисту HTTP, SSL сьогодні дозволяє забезпечити безпеку будь-яких додатків та шифрування TCP/UDP-тунелів.

Технологія SSL (Secure Sockets Layer), розроблена Netscape у 90-х роках, пройшла еволюцію версій (v2 та v3) і була згодом трансформована IETF у стандарт TLS (Transport Layer Security). Основні функції SSL включають автентифікацію сторін через PKI та створення зашифрованих каналів зв’язку [6].

З точки зору моделі OSI, SSL функціонує між транспортним рівнем та рівнем додатків, забезпечуючи шифрування останнього. Процес взаємодії SSL поділяється на чотири етапи:

1. SSL Handshake: узгодження методів шифрування.
2. SSL Change Cipher Spec: генерація та обмін сесійними ключами.
3. SSL Alert: обробка помилок передачі.
4. SSL Record: передача даних.

OpenVPN використовує бібліотеку OpenSSL для криптографічного захисту та автентифікації.

Ключові переваги технології:

- використання перевірених криптографічних алгоритмів (SSL/TLS), що є стандартами IETF;
- простота розгортання та конфігурації;
- підтримка TCP/UDP та можливість роботи через один порт;
- можливість індивідуального налаштування клієнтів;
- гнучкість завдяки використанню інтерфейсів TUN/TAP;
- відсутність конфліктів із NAT;
- висока стійкість до розривів з’єднання, підтримка DynDNS та збереження сесій;
- наявність механізмів стиснення та формування трафіку (traffic shaping);
- висока сумісність із міжмережевими екранами (firewalls);
- ефективність роботи на малопотужному апаратному забезпеченні та відсутність необхідності в root-привілеях (для Linux);
- модульна архітектура.

Обмеження:

- відсутність підтримки спеціалізованого апаратного забезпечення;
- потреба у залученні висококваліфікованих фахівців;
- відсутність вбудованого графічного інтерфейсу (GUI) для адміністративного управління.

1.2.3 Протокол SSH

Одним із протоколів, що дозволяють організувати VPN-з’єднання, є SSH (Secure Shell) - це мережевий протокол сеансового рівня, призначений для віддаленого керування операційними системами та забезпечення тунелювання TCP-з'єднань. На відміну від протоколу Telnet, SSH гарантує конфіденційність шляхом шифрування всього трафіку, включно з даними аутентифікації. Протокол підтримує широкий спектр алгоритмів шифрування та інтегрований у більшість сучасних мережевих ОС [6].

Завдяки здатності інкапсулювати різноманітні мережеві протоколи, SSH дозволяє безпечно передавати дані (наприклад, аудіо- чи відеопотоки) у незахищеному середовищі. Додатковою перевагою є можливість опціонального

стиснення даних перед їх шифруванням.

Стандартна реалізація SSH передбачає використання TCP-порту 22. Процес аутентифікації сервера базується на алгоритмах RSA або DSA. Для ідентифікації клієнта застосовуються аналогічні методи електронно-цифрового підпису, парольна аутентифікація або перевірка IP-адреси (останній метод вважається менш безпечним). Для формування сеансового ключа використовується алгоритм Діффі-Хеллмана, а для захисту цілісності даних - протоколи HMAC-SHA1 або HMAC-MD5 [6].

Технологія SSH-тунелювання дозволяє шифрувати незахищений трафік на етапі його проходження крізь SSH-з'єднання, що забезпечує високий рівень безпеки передачі даних у мережі Інтернет. Практична реалізація включає використання спеціалізованих додатків, створення повноцінних VPN-тунелів або налаштування перенаправлення (port forwarding) TCP-портів [6].

Ключовими перевагами використання SSH для VPN є відсутність необхідності в розгортанні додаткового програмного забезпечення та простота конфігурації. Попри певні обмеження продуктивності, SSH часто перевершує протокол IPsec за параметрами масштабованості та зручності адміністрування, що робить його ефективним рішенням для побудови захищених мережевих каналів.

1.3 Система моніторингу корпоративних мереж

Для забезпечення стабільного функціонування корпоративної мережі критично важливо впровадити ефективну систему моніторингу.

Моніторинг мережі - це процес безперервного контролю стану IT-інфраструктури, спрямований на виявлення низькоефективних або несправних вузлів. У разі виникнення інцидентів система оперативно інформує мережевого адміністратора через встановлені канали зв'язку. Ця діяльність є ключовим компонентом комплексного управління мережею [7].

На відміну від систем виявлення вторгнень, що фокусуються на зовнішніх загрозах, моніторинг мережі забезпечує контроль внутрішніх процесів, зокрема виявляє перевантаження серверів, вихід з ладу мережевого обладнання або розриви з'єднань. Наприклад, для перевірки доступності

вебсервісів здійснюються регулярні HTTP-запити, а для поштових серверів - контрольні сесії протоколів SMTP, IMAP або POP3 [7].

У разі негативних результатів перевірки (тайм-аут, відсутність відповіді, помилка доставки) система моніторингу ініціює відповідні реакції:

- сповіщення адміністративного персоналу;

- автоматичний запуск механізмів відмовостійкості (failover), зокрема перемикання на резервний сервер до моменту усунення несправності.

Окрім діагностики критичних збоїв, моніторинг дозволяє оцінювати загальний стан інфраструктури. Аналіз трафіку мережевих інтерфейсів сприяє оптимізації витрат на послуги провайдерів, що є особливо актуальним в умовах обмеженого вибору інтернет-операторів у бізнес-центрах. Моніторинг ресурсних показників (навантаження на процесор, використання RAM) дозволяє обґрунтовано планувати модернізацію серверного обладнання [7].

Особливої уваги потребує контроль процесів резервного копіювання даних. Моніторинг наявності вільного простору на файлових серверах та успішності створення резервних копій є необхідним заходом для мінімізації ризиків втрати критично важливої корпоративної інформації.

Впровадження систем моніторингу серверів, маршрутизаторів та іншого мережевого обладнання є дієвим інструментом підвищення відмовостійкості IT-інфраструктури та забезпечення оперативного реагування на технічні неполадки.

Попри складність проєктування та налаштування, використання таких рішень є стратегічно доцільним. Через специфіку вже функціонуючих корпоративних мереж часто виникає потреба у виборі та комбінуванні кількох комплексних продуктів для досягнення оптимальних показників надійності.

1.4 Особливості та функції корпоративної мережі

Однією з ключових вимог до виконання кваліфікаційної роботи є забезпечення практичного спрямування результатів шляхом їх апробації на діючому підприємстві. Реалізація роботи в реальних виробничих умовах дозволяє об’єктивно оцінити ефективність мережевих технологій на основі кількісних показників продуктивності каналів зв’язку.

Короткий огляд предметної області: діяльність компанії зосереджена на

реалізації сантехніки та супутніх матеріалів. Структура підприємства включає офіс, складське приміщення та мережу магазинів, територіально розподілених у межах міста. IT-інфраструктура (сервери терміналів, база даних MS SQL та маршрутизатор з налаштуваннями NAT через iptables) винесена на майданчик провайдера.

Централізована модель адміністрування, реалізована через термінальні сервери (RDP), забезпечує високий рівень безпеки та спрощує обслуговування IT-активів. З огляду на роботу з конфіденційною інформацією, питання захисту даних під час їх передачі між філіями є пріоритетним. Взаємодія магазинів із серверами здійснюється виключно через віддалені ресурси. Наразі інфраструктура підприємства базується на маршрутизаторах. Серверна частина (термінали, MS SQL) функціонує під управлінням MS Windows 2025.

Для створення системи, що відповідає встановленим вимогам та враховує наявні обмеження, визначено такі завдання:

1. Побудова захищеної корпоративної мережі підприємства.

2. Організація безпечних каналів зв’язку між магазинами та серверним сегментом.

3. Аналіз продуктивності розгорнутих каналів передачі даних.

4. Впровадження системи моніторингу мережевого обладнання та оповіщення про збої.

Структуру спроєктованої корпоративної мережі підприємства наведено на рис. 1.4.

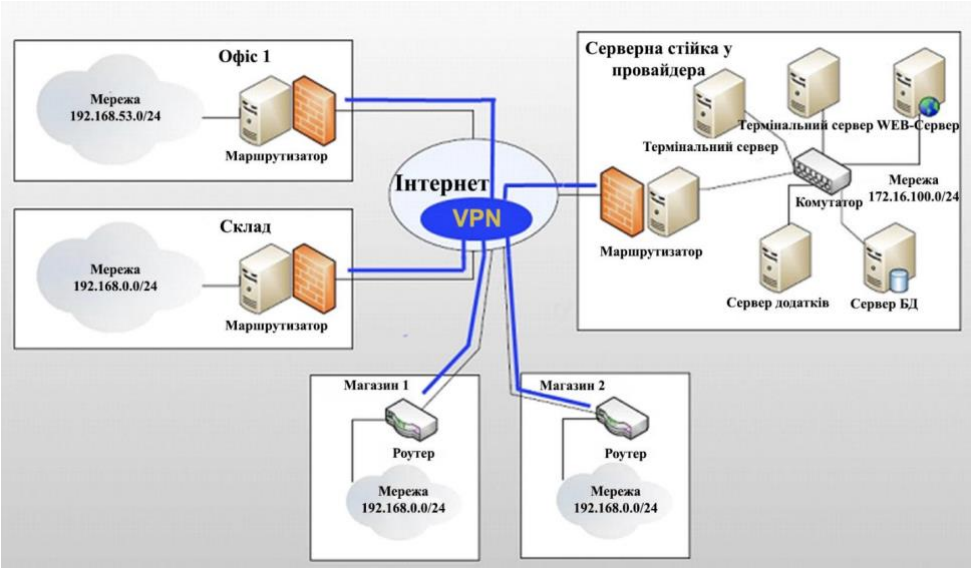


Рисунок 1.4 - Проектована корпоративна мережа

2 ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ VPN У КОРПОРАТИВНІЙ МЕРЕЖІ ПІДПРИЄМСТВА

2.1 Впровадження технології OpenVPN

Першим етапом реалізації проєкту є об’єднання офісу, складських приміщень та серверної інфраструктури в єдину корпоративну мережу. З метою оптимізації архітектури зв’язок буде забезпечено шляхом створення захищених тунельних з’єднань безпосередньо між маршрутизаторами, що виключає потребу в індивідуальному підключенні кінцевих робочих станцій.

Система базується на трьох маршрутизаторах під керуванням ОС OPNsense. Маршрутизація трафіку між внутрішніми сегментами мережі та зовнішніми ресурсами реалізується з використанням технології NAT та правил iptables.

Для структурування мережевої конфігурації встановлено наступну номенклатуру вузлів та їх мережеві параметри (табл. 2.1):

- Office1: офісний маршрутизатор;
- Sklad: маршрутизатор складського комплексу;
- Prov: маршрутизатор серверної інфраструктури (провайдер);
- M1: маршрутизатор магазину №1;
- M2: маршрутизатор магазину №2.

Таблиця 2.1 - Мережеві параметри маршрутизаторів

Office1:

Мережа	Інтерфейс	IP-адреса	Маска	Шлюз
Інтернет	eth2	213.182.175.230	255.255.255.252	213.182.175.229
Локальна	eth1	192.168.53.250	255.255.255.0	-

Sklad:

Мережа	Інтерфейс	IP-адреса	Маска	Шлюз
Інтернет	eth2	79.142.87.206	255.255.255.252	79.142.87.211
Локальна	eth1	192.168.0.1	255.255.255.0	-

Prov:

Мережа	Інтерфейс	IP-адреса	Маска	Шлюз
Інтернет	eth2	195.2.240.68	255.255.255.252	195.2.240.60
Локальна	eth1	172.16.100.8	255.255.255.0	-

Апаратне налаштування маршрутизаторів у філіях не є критично важливим, тому цей етап ми пропустимо.

Перейдемо безпосередньо до процесу встановлення та конфігурації.

OPNsense - це дистрибутив Linux, заснована на ядрі Linux. Розробка OpenWrt стала можливою завдяки використанню виробником програмного забезпечення ліцензії GNU GPL, яка вимагає від розробників публікувати всі похідні продукти під тією ж ліцензією. Для керування пакунками та встановлення оновлень система використовує менеджер yum. Всі роботи з налаштування та адміністрування виконуються дистанційно за допомогою OpenSSH-сервера на маршрутизаторах та клієнта PuTTY [8].

Першочергово налаштуємо маршрутизатор Prov, який виконуватиме функцію OpenVPN-сервера. Оскільки пакет OpenVPN відсутній у стандартному репозиторії, необхідно підключити додатковий репозиторій rpmforge:

```
Prov> rpm -Uhv
http://apt.sw.be/redhat/el5/en/x86_64/rpmforge/RPMS//rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

Ця команда завантажує пакет RPM із репозиторію та ініціює його встановлення. Після того, як пакет OpenVPN стане доступним у системі, виконуємо команду для його встановлення:

```
Prov> yum install openvpn
```

Після успішного встановлення OpenVPN необхідно згенерувати кореневий сертифікат сервера, сертифікати та ключі клієнтів, сертифікат і ключ сервера, а також TLS-ключ.

Для виконання цих операцій перейдемо до каталогу конфігурації OpenVPN та створіть окремі директорії для зберігання ключів та конфігураційних файлів клієнтів:

```
Prov> cd /etc/openvpn
Prov> mkdir keys
Prov> mkdir ccd
```

Виконується завантаження змінних для генерації ключів у пам'ять, після чого розпочинається процес створення сертифіката авторизації.

Prov>./vars
Prov>./build-ca
Generating a 1024 bit RSA private key
.....++++++
..++++++
writing new private key to 'ca.key'

Буде запропоновано ввести дані для формування запиту на сертифікат. Ці відомості утворюють так зване «розрізнювальне ім’я» (Distinguished Name, DN). Деякі поля є необов'язковими, тому їх можна залишити порожніми. Для певних полів передбачено значення за замовчуванням. Щоб залишити поле порожнім, введіть крапку (.).

Країна
Country Name (2 letter code) [UA]:UA
Область
State or Province Name (full name) [KHO]:KHO
Місто
Locality Name (eg, city) [Kharkiv]:KH
Організація
Oganization Name (eg, company) []:server
Підрозділ
Organizational Unit Name (eg, section) []:server
Ім’я сервера OpenVPN
Common Name (eg, your name or your server's hostname) []:server
Name []:server
Email Address [me@myhost.mydomain]:

Формуємо сертифікат X.509 для створеного сервера:

```
Prov> ./build-key-server server
# Країна
Country Name (2 letter code) [UA]:UA
# Область
State or Province Name (full name) [KHO]:KHO
# Місто
Locality Name (eg, city) [Kharkiv]:KH
# Організація
Oganization Name (eg, company) [Provider]:server
# Підрозділ
Organizational Unit Name (eg, section) [Provider]:server
# Ім’я сервера OpenVPN
Common Name (eg, your name or your server's hostname) [Provider]:server
Name []:server
# Поштова адреса
Email Address [root@localhost]:
Please enter the following 'extra' attributes
to be sent with your certificate request
# Пароль
A challenge password []:987654321
# Назва організації
An optional company name []:server
```

Після появи запиту на підписання сертифіката підтверджуємо дію.
Для створення ключа доступу для об’єкта «Office1» виконаємо команду:

```
Prov>./build-key-server office
Generating a 1024 bit RSA private key
.........++++++
.......++++++
writing new private key to 'client.key'
```

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [UA]:UA
State or Province Name (full name) [KHO]:KHO
Locality Name (eg, city) [Kharkiv]:KH
Oganization Name (eg, company) [server]: company
Organizational Unit Name (eg, section) [Office1]:office
Common Name (eg, your name or your server's hostname) [Office1]: office

Email Address [root@localhost]:
Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:987654321
An optional company name []:office

Аналогічним чином генеруємо ключі для складського приміщення та торгових точок. Для налаштування обміну даними через незахищені канали необхідно згенерувати ключ Діффі-Гелмана:

Prov>./build-dh

Для забезпечення TLS-автентифікації згенеруйте ключ за допомогою команди:

Prov>openvpn --genkey --secret keys/ta.key

Після виконання вказаних операцій у каталозі keys/ будуть сформовані такі файли:

- ca.crt - головний сертифікат CA (необхідний для сервера та клієнтів);

- dh1024.pem - ключ Діффі-Гелмана (тільки для сервера);

- server.crt - сертифікат сервера (тільки для сервера);

- server.key - приватний ключ сервера (конфіденційно, тільки для сервера);

- office1.crt, sklad.crt, m1.crt, m2.crt - клієнтські сертифікати (для відповідних вузлів);

- office1.key, sklad.key, m1.key, m2.key - приватні ключі клієнтів (конфіденційно, для відповідних вузлів);

- ta.key - TLS-ключ (для сервера та клієнтів).

На сервері необхідно розмістити файли: ca.crt, dh1024.pem, server.crt, server.key, ta.key. Клієнтським вузлам передаються файли ca.crt, ta.key та відповідні індивідуальні сертифікати й ключі.

Процес генерації ключів та сертифікатів завершено. Переходимо до етапу конфігурації. Створюємо файл `server.conf` із наступним змістом:

```
# Порт для роботи сервера
port 5000
# Протокол передачі даних
proto udp
# Тип мережевого інтерфейсу та його ідентифікатор
dev tun0
# Шлях до файлу сертифіката центру сертифікації (CA)
ca /etc/openvpn/keys/ca.crt
# Шлях до файлу сертифіката сервера
cert /etc/openvpn/keys/server.crt
# Шлях до файлу закритого ключа сервера
key /usr/local/etc/openvpn/keys/server.key
# Шлях до файлу параметрів Діффі-Хелмана
dh /usr/local/etc/openvpn/keys/dh1024.pem
# IP-адреса та маска підмережі віртуальної мережі
server 10.10.200.0 255.255.255.0
```

Маршрути для передачі клієнтам та відповідна маска підмережі для забезпечення доступу до ресурсів внутрішньої мережі за сервером OpenVPN

```
# Prov
push "route 172.16.100.0 255.255.255.0"
# Office1
push "route 192.168.53.0 255.255.255.0"
# Sklad
push "route 192.168.0.0 255.255.255.0"
# Вказується шлях зберігання файлів з IP-адресами клієнтів
client-config-dir ccd
# додаються маршрути сервер-клієнтів
route 10.10.200.0 255.255.255.0
# Office1
маршрут 192.168.53.0 255.255.255.0
# Sklad
маршрут 192.168.0.0 255.255.255.0
# Дозвіл клієнтам бачити один одного (за віртуальними IP), за замовчуванням клієнти бачать тільки сервер
client-to-client
# Вмикається аутентифікація TLS
tls-server
# Вказується tls-ключ
tls-auth keys/ta.key 0
# Таймаут до повторного підключення
tls-timeout 100
# Вибирається алгоритм хешування
auth MD5
# Вмикається шифрування пакетів
cipher BF-CBC
# Перевіряється активність з'єднання кожні 20 секунд, якщо протягом 100 сек. немає відповіді, з'єднання закривається
keepalive 20 100

# Включається стиснення трафіку
comp-lzo
# Вказується який користувач та група працюватиме OpenVPN
```

```
user nobody
group nobody
# Не перечитувати ключі після отримання SIGUSR1 або ping-restart
persist-key
# Не закривати та не відкривати знову пристрій TUN\TAP після отримання SIGUSR1 або ping-restart
persist-tun
# Реєстрація
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
# Рівень інформації для налагодження
verb 3
```

Підготовка конфігураційних файлів для клієнтів здійснюється шляхом створення відповідних файлів у каталозі /etc/openvpn/ccd на сервері [8]. Для кожного клієнта необхідно створити окремий файл (назва файлу повинна відповідати імені, вказаному в сертифікаті), зокрема: office1, sklad, m1, m2. Файли повинні містити наступну конфігурацію:

```
office1
ifconfig-push 10.10.200.2 10.10.200.1
iroute 192.168.53.0 255.255.255.0
sklad
ifconfig-push 10.10.200.3 10.10.200.1
iroute 192.168.53.0 255.255.255.0
m1
ifconfig-push 10.10.200.4 10.10.200.1
m2
ifconfig-push 10.10.200.5 10.10.200.1
```

Згідно з налаштуваннями, для клієнтів із відповідними сертифікатами призначено віртуальні IP-адреси та шлюз 10.10.200.1, а також встановлено маршрутизацію трафіку через тунель до мереж клієнтів. Маршрутизація для мереж магазинів не налаштовується, оскільки це не входить у технічне

завдання.

Після завершення налаштування сервера необхідно ініціювати запуск OpenVPN:

```
Prov> service openvpn start
```

У разі коректного виконання конфігурації має відобразитися віртуальний tun-пристрій:

```
Prov> ifconfig tun0
tun0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:10.10.200.1 P-t-P:10.10.200.2 Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500
Metric:1
RX packets:9 errors:0 dropped:0 overruns:0 frame:0
TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
```

Якщо пристрій не відображається у списку, це свідчить про наявність помилок у файлах конфігурації. Необхідно проаналізувати лог-файли, усунути виявлені неточності та повторно запустити процес.

Переходимо до налаштування клієнтських станцій. Оскільки всі конфігураційні файли ідентичні, розглянемо налаштування одного з них. На маршрутизаторах office1 та sklad слід встановити OpenVPN за аналогією до налаштування сервера.

Створимо конфігураційний файл client.conf:

```
dev tun
proto udp
remote 195.2.240.68 #(реальний IP-адреса сервера)
port 5000
client
resolv-retry infinite
```

ca keys/ca.crt
cert keys/client.crt
key keys/client.key
tls-client
tls-auth keys/ta.key 1
auth MD5
cipher BF-CBC
ns-cert-type server
comp-lzo
persist-key
persist-tun
Додавання маршруту до мережі за сервером. Цей рядок не є обов'язковим для конфігураційних файлів магазинів.
up /etc/openvpn_up.sh
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
verb 3

Формуємо скрипт openvpn_up.sh для автоматизації маршрутизації:

#!/bin/sh
/sbin/route add -net 172.16.100.0 netmask 255.255.255.0 gw 10.10.200.1
tun0

На цьому етапі налаштування OpenVPN завершено. Необхідно скопіювати відповідні файли на вузли «office1» та «sklad», після чого запустити службу OpenVPN. У разі виникнення помилок під час запуску слід звернутися до лог-файлів.

Крім того, для забезпечення повноцінного функціонування системи необхідно налаштувати трансляцію мережевих адрес (NAT). Це дозволить пакетам, що надходять від клієнтських пристроїв на сервер, коректно виходити в мережу Інтернет та повертатися до адресатів.

Prov> iptables —t nat —A POSTROUTING —s 10.10.200.0/24 —o eth1 —

j MASQUERADE

Наразі три сегменти мережі інтегровані та знаходяться у стані взаємодії. Наступним етапом є налаштування підключення терміналів у магазинах до центральних серверів. Оскільки на робочих станціях у магазинах використовується ОС Windows 2011, необхідно завантажити дистрибутив OpenVPN з офіційного ресурсу та виконати його інсталяцію. Після завершення встановлення слід розмістити ключі доступу та файл конфігурації «m1» у директорії «config», після чого ініціювати запуск служби.

На цьому етапі процес конфігурації завершено. Створення захищеної корпоративної мережі дозволяє здійснювати пряме підключення до серверів. Перевірку коректності шифрування можна виконати шляхом аналізу трафіку на одному з маршрутизаторів за допомогою утиліти TCPDUMP [9].

Приклад відображення нешифрованого трафіку:

18:27:15.752295 IP cl230-175-182-213.cl.metrocom.ru.40887 > 195.2.240.68.ssh: . 2826496:2827944(1448) ack 1009 win 10080 <nop,nop,timestamp 2791385847 256970382>

18:27:15.752347 IP 195.2.240.68.ssh > cl230-175-182-213.cl.metrocom.com.40887: . ack 2783056 win 65535 <nop,nop,timestamp 256970382 2791385774,nop,nop,sack 1 {2785952:2827944}>

18:27:15.755042 IP cl230-175-182-213.cl.metrocom.ru.40887 > 195.2.240.68.ssh: 2827944:2829392(1448) ack 1009 win 10080 <nop,nop,timestamp 2791385850 256970382>

18:27:15.755096 IP 195.2.240.68.ssh > cl230-175-182-213.cl.metrocom.com.40887: . ack 2783056 win 65535 <nop,nop,timestamp 256970382 2791385774,nop,nop,sack 1 {2785952:2829392}>

Приведемо також приклад зашифрованого трафіку:

18:24:18.247960 IP 195.2.240.68.sieve > cl230-175-182-

213.cl.metrocom.ru.sieve: UDP, length 113
18:24:18.248040 IP 195.2.240.68.sieve > cl230-175-182-
213.cl.metrocom.com.sieve: UDP, length 113
18:24:18.250915 IP cl230-175-182-213.cl.metrocom.ru.sieve >
195.2.240.68.sieve: UDP, length 1441
18:24:18.251291 IP 195.2.240.68.sieve > cl230-175-182-
213.cl.metrocom.com.sieve: UDP, length 113

2.2 Реалізація на основі технології SSH

2.2.1 Формування SSH-тунелю

У разі функціонування офісу, складу та магазинів (інформація про мережеву структуру наведена у п. 2.1) виключно на базі серверів у термінальному режимі, доцільно застосувати спрощений та оперативний метод організації VPN-тунелю. Використання пакета OpenSSH на програмних маршрутизаторах дозволяє створити зашифрований канал передачі даних із використанням єдиного цільового порту. Для взаємодії з термінальним сервером клієнтські станції використовують протокол RDP (порт 3389). Схему зазначеної конфігурації представлено на рисунку 2.1, де синім кольором виділено SSH-тунель, призначений для передачі трафіку протоколу RDP [9].

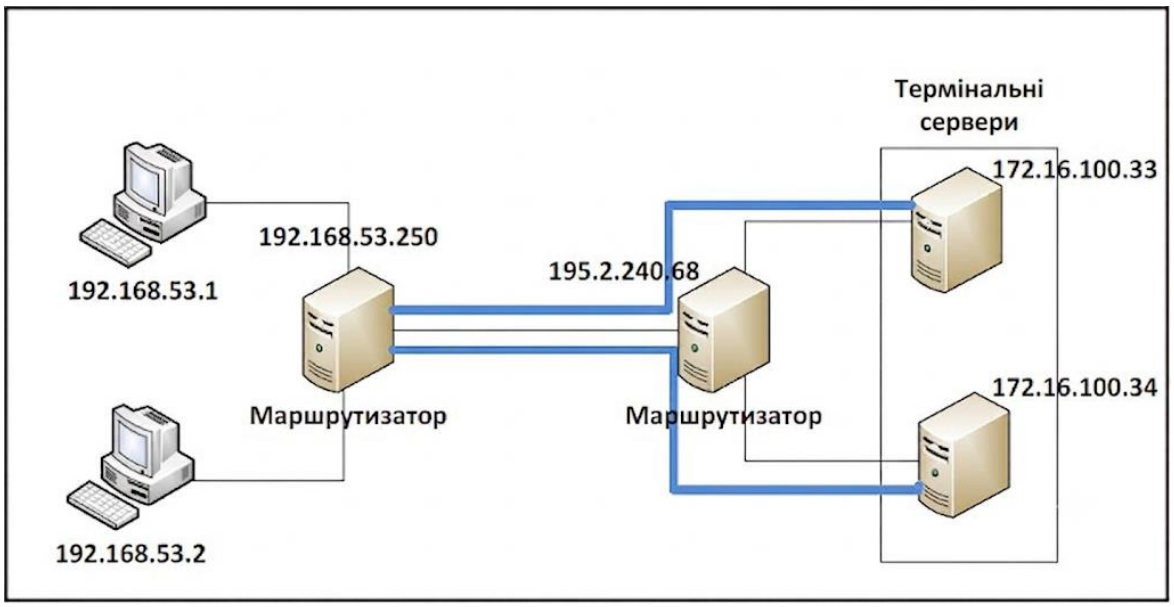


Рисунок 2.1 - Використання SSH-тунелів

Реалізація, як зазначалося раніше, є досить простою, проте OpenSSH має певний недолік: у разі тривалого простою каналу відбувається його розрив. Цю проблему можна вирішити за допомогою утиліти AutoSSH, яка забезпечує підтримку з'єднання шляхом надсилання heartbeat-пакетів через додатковий порт під час відсутності активності [9]. Нижче наведено інструкцію зі створення двох каналів до термінальних серверів:

```
Prov> export AUTOSSH_DEBUG=1
Prov> export AUTOSSH_GATETIME=0
Prov> export AUTOSSH_PORT=20037
Prov> autossh -f -N -g -l root -L 1002:172.16.100.33:3389 195.2.240.68
Prov> export AUTOSSH_PORT=20040
Prov> autossh -f -N -g -l root -L 1002:172.16.100.34:3389 195.2.240.68
```

Розберемо конкретніше команди:

```
Prov> autossh -f -N –g –c aes256 -l root -L 1002:172.6.100.33:3389
195.2.240.68
```

Параметри конфігурації:

-f: переведення процесу у фоновий режим без прямого підключення до хоста;

-N: заборона виконання віддалених команд;

-g: надання дозволу на підключення віддаленим хостам;

-c: активація шифрування;

-l: визначення імені користувача для авторизації на сервері;

-L: конфігурація локального порту для перенаправлення трафіку до цільового сервера та порту.

Налаштування зашифрованого тунелю завершено, система готова до роботи. Для створення додаткового тунелю необхідно призначити новий порт для передачі пакетів heartbeat за допомогою команди export AUTOSSH_PORT=20050, після чого повторно виконати команду ініціалізації SSH-тунелю із зазначенням відповідного порту.

2.2.2 Створення захищеного тунелю SSH VPN

Будемо відтворювати ту саму архітектуру корпоративної мережі, що була реалізована за допомогою OpenVPN, але тепер з використанням функціоналу OpenSSH, інтегрованого в Linux. Для демонстрації буде достатньо конфігурації для з'єднання двох мереж, оскільки розширення до більшої кількості мереж передбачає повторення аналогічних кроків.

Починаючи з версії 4.3, OpenSSH підтримує створення зашифрованих тунелів через пристрої tun/tap. Ця можливість є аналогічною до OpenVPN, який базується на протоколі TLS [9].

Зашифрований тунель формується на основі одного TCP-з'єднання, що забезпечує ефективне та просте розгортання VPN на IP-рівні [9].

Першим кроком є модифікація конфігураційного файлу OpenSSH (/etc/ssh/sshd_config) для надання дозволу на створення пристроїв tun/tap та виконання з правами root [9]. Необхідно включити наступні параметри:

```
PermitRootLogin yes
PermitTunnel yes
```

Щоб створити безпечний VPN-тунель між офісною мережею (192.168.53.0/24) та мережею провайдера (172.16.100.0/24), потрібно виконати таку послідовність дій:

1. Здійснити SSH-підключення з маршрутизатора офісу до маршрутизатора колокейшну, використовуючи опцію -w для ініціації тунелю.
2. Налаштувати IP-адреси для цього SSH-тунелю, що є одноразовою процедурою як для клієнтської, так і для серверної сторони.
3. Додати необхідні маршрути, щоб забезпечити зв'язок між обома мережами.
4. За потреби, увімкнути функцію NAT на внутрішньому інтерфейсі шлюзу.

Процес починається з маршрутизатора офісу, але основні команди конфігурації виконуються на маршрутизаторі провайдеру. Параметри -w 0:0 вказують на створення віртуальних пристроїв tun0 на обох кінцях тунелю, а -c та -C активують шифрування та стиснення даних відповідно.

Office1> ssh –c aes256 -C –w0:0 root@195.2.240.68

Наступні команди виконуються на маршрутизаторі мережі Prov. Задаємо IP-адресу та маску підмережі:

Prov> ifconfig tun0 10.0.1.1 netmask 255.255.255.252

Додаємо маршрут до мережі підприємства office1:

Prov> route add -net 192.168.53.0 netmask 255.255.255.0 dev tun0

Вмикаємо NAT:

Prov> echo 1 > /proc/sys/net/ipv4/ip_forward

Створюємо правило для перенаправлення пакетів з VPN у глобальну:

Prov> iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

Налаштовуємо маршрутизатор office1:

Office1> ifconfig tun0 10.0.1.2 netmask 255.255.255.252
Office1> route add -net 172.16.100.0 netmask 255.255.255.0 dev tun0
Office1> echo 1 > /proc/sys/net/ipv4/ip_forward
Office1> iptables -t nat -A POSTROUTING -o eth1 -j

На цьому етапі конфігурація завершена, і VPN-мережа успішно розгорнута. Для підключення окремих комп'ютерів під керуванням Windows 2011 використовується SSH-клієнт Putty.

2.3 Оцінка продуктивності сформованої корпоративної мережі

У попередніх підрозділах було розглянуто дві технології – OpenVPN та SSH – для створення захищених корпоративних мереж за допомогою VPN.

Наразі OpenVPN є провідною технологією на ринку захищених мереж, тоді як SSH-тунелювання лише починає набирати популярність. Щоб визначити, яка технологія найкраще підходить для конкретних вимог корпоративної мережі в реальних умовах, необхідно оцінити їхню продуктивність та обґрунтувати їхні сильні та слабкі сторони.

Почнемо з оцінки продуктивності захищених каналів. На рис. 1.4 зображено розгорнуту корпоративну мережу, продуктивність захищених каналів якої буде виміряна за допомогою програми iPerf. Клієнтська частина генеруватиме трафік і надсилатиме його на серверну частину. Після отримання даних буде сформовано звіт про швидкість передачі даних.

2.3.1 Оцінка продуктивності мережі з технологією OpenVPN

Для візуалізації показників продуктивності каналу, сформованого за допомогою OpenVPN, будуть використані результати вимірювань, представлені в Додатку А. На рис. 2.2 наведено графіки значень RTT (Round-Trip Time). Аналіз цих графіків демонструє, що відмінності у часі відгуку між прямим з'єднанням та з'єднанням через VPN є незначними. Крім того, активація функції стиснення даних не виявляє суттєвого впливу на показники RTT.

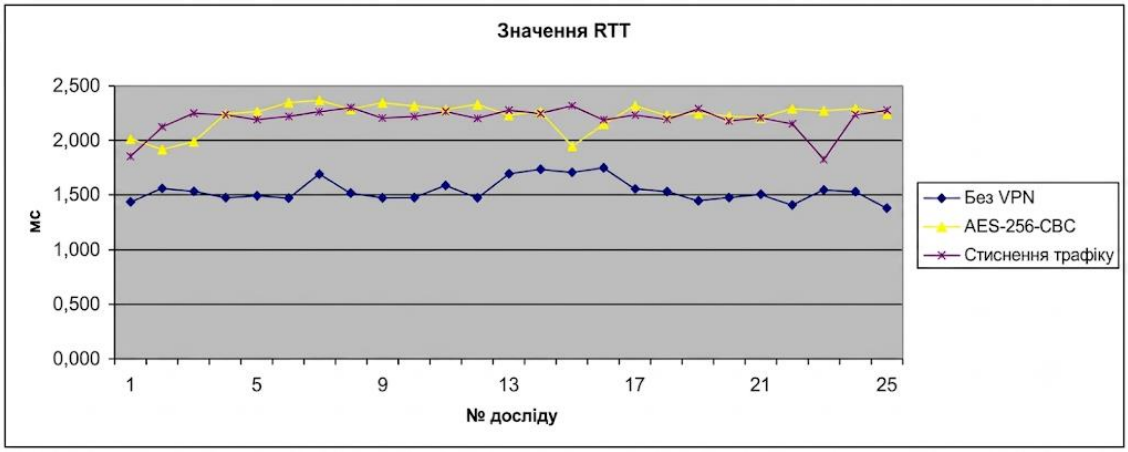


Рисунок 2.2 - Графіки значень RTT

Графіки пропускної здатності каналу, представлені на рис. 2.3, свідчать про наступне: використання VPN з шифруванням AES-256-CBC призводить до незначного зниження швидкості на 0,5 Мбіт/с, що становить 5,1% від вихідної пропускної здатності; застосування стиснення для шифрованого трафіку

демонструє значний позитивний ефект, збільшуючи швидкість на 3 Мбіт/с (32,9%).

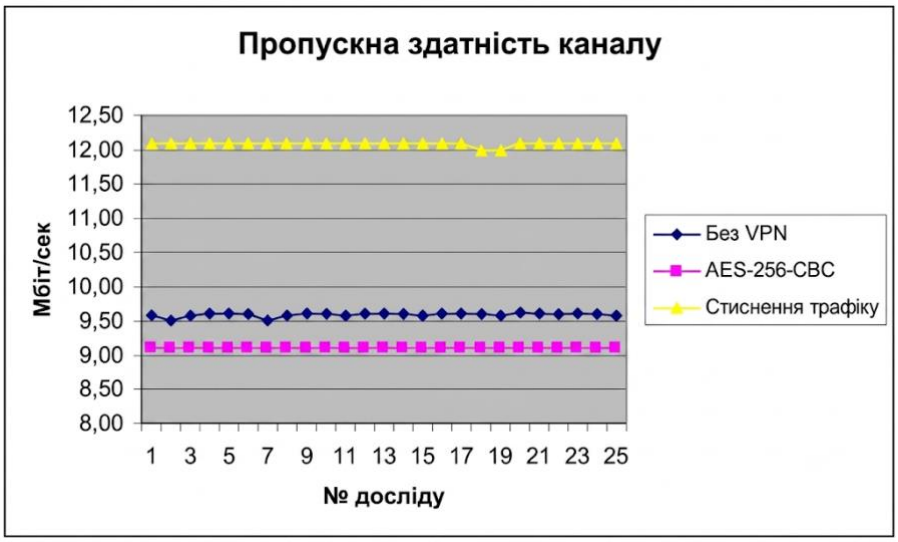


Рисунок 2.3 - Графіки пропускної здатності каналу

Для порівняння, рис. 2.4 показує вплив цих налаштувань (шифрування та стиснення) на завантаження центрального процесора маршрутизаторів при роботі з OpenVPN.

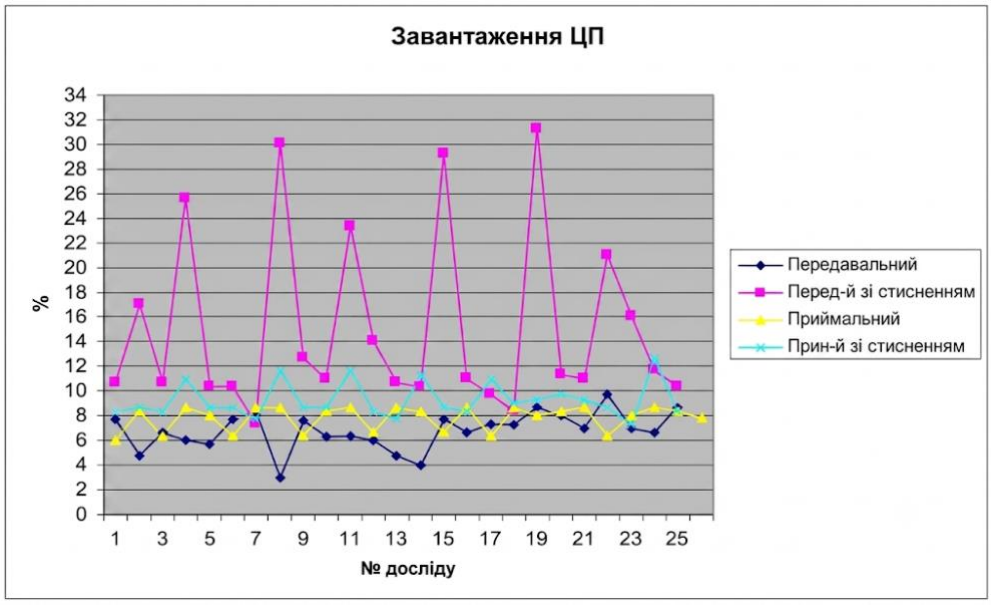


Рисунок 2.4 - Графіки завантаження ЦП

Як і очікувалося, найвищий рівень навантаження на систему спостерігається при шифруванні трафіку з одночасним стисненням, що становить 14,992%.

На основі отриманих графіків можна зробити наступні висновки щодо продуктивності VPN-каналів, реалізованих за допомогою OpenVPN:

1. Навантаження на центральний процесор (ЦП): В умовах даного експерименту, де маршрутизатор виконує лише функції VPN-сервера і не задіяний в інших ресурсомістких процесах, показник завантаження ЦП є незначним.

2. Час відгуку (RTT): Різниця в часі відгуку між з'єднанням без VPN та з VPN виявилася мінімальною (лише 0,5 мс), що робить цей критерій також несуттєвим для оцінки.

3. Пропускна здатність: Графіки демонструють зниження швидкості передачі даних при використанні VPN приблизно на 0,5 Мбіт/с в середньому. На даний момент це падіння не є критичним, враховуючи високі швидкості, що надаються сучасними інтернет-провайдерами.

4. Ефект від стиснення трафіку: Застосування стиснення трафіку призводить до помітного збільшення пропускної здатності каналу – на 3 Мбіт/с. Хоча це й спричиняє зростання навантаження на ЦП, як зазначалося раніше, це не є суттєвим обмеженням.

Таким чином створення захищеної корпоративної мережі за допомогою OpenVPN дозволяє об'єднати кілька офісів в єдиний простір з надійним шифруванням даних. Додатково, стиснення трафіку сприяє підвищенню швидкості передачі інформації, що полегшує обмін даними. Технологія OpenVPN повністю виправдовує себе, підвищуючи ефективність роботи з інформацією в мережі. Серед недоліків можна виділити певну складність у налаштуванні та розгортанні VPN-мережі. Однак, значним плюсом є її кросплатформенність.

2.3.2 Оцінка продуктивності мережі з технологією SSH

Для аналізу продуктивності каналу, створеного за допомогою SSH, будуть використані дані, зібрані під час тестування, представлені в Додатку Б.

На діаграмах, що ілюструють рис. 2.5, відображено динаміку часу відгуку SSH-з'єднань. Зафіксоване середнє збільшення цього показника на 0,8 мс не

розглядається як критичне і не впливає на продуктивність навіть найбільш чутливих до затримок додатків.

Аналіз графіків пропускної здатності каналу, представлених на рис. 2.6-2.7, демонструє результати, порівнянні з показниками, отриманими при використанні OpenVPN.

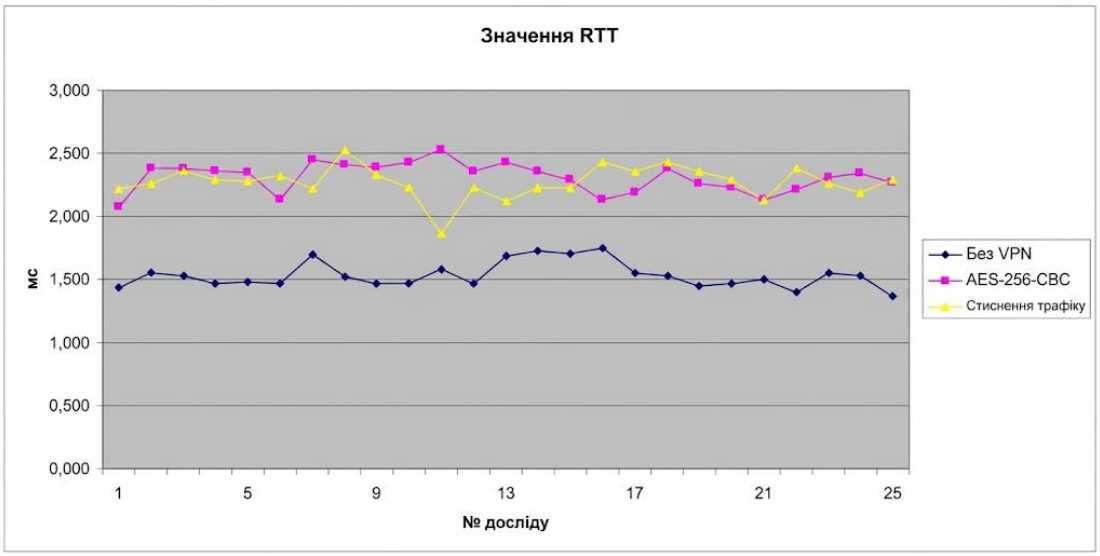


Рисунок 2.5 - Графіки значень RTT

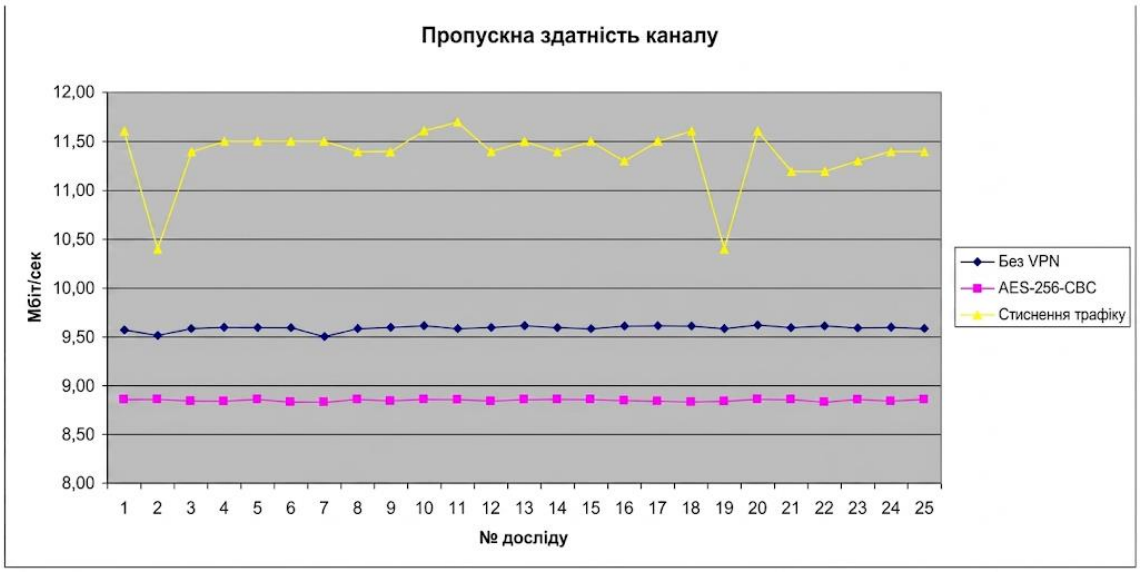


Рисунок 2.6 - Графіки пропускної здатності каналу

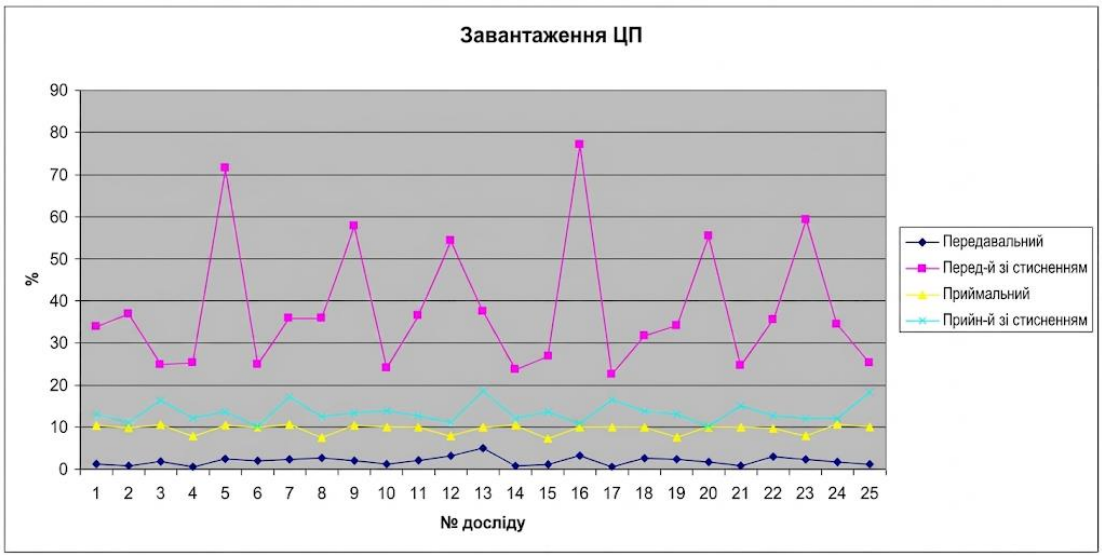


Рисунок 2.7 - Графіки завантаження ЦП

Рисунок 2.7 ілюструє динаміку завантаження центрального процесора при використанні SSH з шифруванням, як зі стисненням трафіку, так і без нього. Активація шифрування викликає помітне зростання навантаження на ЦП, досягаючи середнього значення 37,9%. Хоча це число є досить високим, воно не становить критичної проблеми.

З аналізу представлених даних можна зробити висновок, що застосування стиснення трафіку в SSH призводить до підвищення продуктивності приблизно на 25%, але водночас суттєво збільшує навантаження на ЦП. Серед позитивних аспектів цієї технології варто виділити можливість гнучкого налаштування SSH-тунелів через специфічні порти. Це може бути корисним для створення, наприклад, автоматизованих систем резервного каналу зв'язку, що активуються у разі відсутності зв'язку з одним із маршрутизаторів. Крім того, слід підкреслити простоту розгортання та конфігурації VPN-мереж, їхню сумісність з різними операційними системами, високий рівень надійності та легкість масштабування.

2.3.3 Порівняння SSH та OpenVPN

Сформуємо порівняльну таблицю на основі отриманих експериментальних даних (табл. 2.2).

Таблиця 2.2 – Порівняльна характеристика протоколів VPN

Параметр	OpenVPN	SSH
Складність створення	Досить складне налаштування. Складність налаштування маршрутизації для декількох мереж, можливі труднощі з брандмауером.	Легке налаштування, що не вимагає особливих знань. На все створення йде не більше 10 хвилин.
Масштабованість	Підключення ще однієї мережі або клієнта вимагає зміни конфігураційних файлів на сервері та додавання їх на клієнт. Завдяки OpenVPN клієнт під Windows має перевагу перед SSH. Не вимагає навчання персоналу.	Для підключення ще однієї мережі потрібно повторити ті самі дії, що й при об'єднанні попередніх. Для підключення комп'ютерів під Windows потрібно одноразове навчання персоналу.
Продуктивність	Завдяки стисненню можна досягти відмінних результатів, що перевищують вихідне з'єднання. Стиснення відбувається вибірково, тобто те, що стиснути не можна, пропускається. Це зменшує навантаження на ЦП.	Продуктивність трохи нижча, ніж у OpenVPN. Стискається весь трафік – велике навантаження на ЦП.
Завантаження ЦП	У середньому не більше 15%	В середньому не більше 38%
Кросплатформеність	Так	Так
RTT	На 0,5 мс більше, ніж у вихідного з'єднання	На 0,8 мс більше, ніж у вихідного з'єднання
Документація	Багато документації на офіційному сайті. Численні форуми та обговорення.	Документація є у вбудованому довіднику. Інформації щодо налаштування поки що мало.
Додаткові можливості	-	Створення ssh-тунелів.
Впровадження в існуючу мережу	Можуть бути проблеми з налаштуванням firewall’а.	Легке впровадження.
Захищеність	Шифрування 256-бітним AES-ключем	Шифрування 256-бітним ключем AES
Поширеність	Провідна технологія створення VPN-мереж.	Сам протокол SSH існує вже дуже давно, але створення SSH-VPN-мереж на сьогоднішній день зустрічається рідко.

Зваживши всі переваги та недоліки, найбільш раціональним рішенням є інтеграція обох технологій. Зокрема, доцільно застосувати SSH-тунелювання для забезпечення безпечного доступу та OpenVPN для створення повноцінних VPN-мереж.

3 ФОРМУВАННЯ СИСТЕМИ МОНІТОРИНГУ КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Система моніторингу серверної інфраструктури Nagios

Nagios є гнучким інструментом для моніторингу серверної інфраструктури, що особливо ефективний при роботі з великою кількістю серверів. Це програмне забезпечення з відкритим кодом, призначене для спостереження та контролю стану комп'ютерних систем та мереж. Його основна функція – моніторинг обчислювальних вузлів та служб, з автоматичним сповіщенням адміністратора про будь-які зміни в їх статусі (припинення або відновлення роботи). У рамках даного проекту, ми будемо використовувати Nagios для моніторингу маршрутизаторів, серверів та процесів, що на них запущені. Для спрощення процесу налаштування, Nagios буде встановлено на маршрутизатор colo [10].

Система Nagios надає як засоби для збору даних моніторингу, так і веб-інтерфейс для управління та візуалізації поточного стану серверів. Тому, крім встановлення самої програми Nagios, буде потрібно розгорнути веб-сервер. Почнемо з встановлення необхідних компонентів [10].

Встановлюємо компілятор мови C++, його бібліотеки та інструменти для роботи з графічними елементами.

```
Prov> yum install gcc glibc glibc-common gd gd-devel
```

Встановлюємо веб-сервер Apache, формуємо користувачів та групу з якими працюватиме nagios. Також встановлюємо пароль.

```
Prov> yum install httpd
Prov> useradd -m nagios
Prov> groupadd nagcmd
Prov> passwd nagios
```

Формуємо одну групу для nagios та apache, щоб зменшити проблеми з правами при запуску скриптів на веб-сервері

Prov> /usr/sbin/usermod -a -G nagcmd nagios
Prov> /usr/sbin/usermod -a -G nagcmd apache

Встановлюємо останню версію системи та необхідні модулі:

Prov> wget http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz
Prov> wget
Prov> tar xzf nagios-3.2.1.tar.gz
Prov> cd nagios-3.2.1
Prov>./configure
Prov> make all
Prov> make install
Prov> make install-init
Prov> make install-config
Prov> make install-commandmode

Записуємо налаштування в Apache, та повторюємо всі дії для додаткових модулів

Prov> make install-webconf
Prov> tar xzf nagios-plugins-1.4.11.tar.gz
Prov> cd nagios-plugins-1.4.11
Prov>./configure
Prov> make
Prov> make install

Встановлюємо пароль для веб-інтерфейсу через утиліту htpasswd та перевіряємо правильність конфігурації

Prov> htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
Prov> /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Запускаємо Nagios і Apache при відсутнеості помилок

Prov> service nagios start
Prov> service httpd restart

Переконатися, що система працює, можна, зайшовши у веб-інтерфейс за адресою: http://ip.адреса.сервера/nagios. У нашому випадку IP-адреса сервера – 195.2.240.68. Якщо сервер Apache запущений, а шлях до сайту Nagios у конфігураційному файлі вказаний коректно, відкриється сторінка авторизації. Після введення логіна та пароля відбудеться перехід до вітальної сторінки Nagios (рис. 3.1). Якщо цього не відбувається, варто перевірити помилки у логах Apache, які знаходяться у каталозі /var/log/httpd [10].



Рисунок 3.1 - Сторінка привітання Nagios

Розглянемо навігаційне меню:

– Home – головна сторінка з вітанням, де можна дізнатися про доступні оновлення системи;

– Documentation – великий обсяг документації щодо налаштування системи;

– Tactical Overview – короткий огляд інформації про об’єкти, що моніторяться;

– Map – відображення мережевої карти, де активні сервери позначені зеленим, а неактивні – червоним (рис. 3.2);

– Hosts – показує стан кожного окремого об’єкта моніторингу;

– Services – демонструє стан запущених процесів на серверах, а також такі параметри, як RTT, вільне місце на диску, завантаження процесора та пам’яті (рис. 3.3);

– Event log – звіти системи моніторингу.

Інші пункти меню є варіаціями наведених вище, що забезпечують зручний перегляд інформації, наприклад, з розподілом за групами.

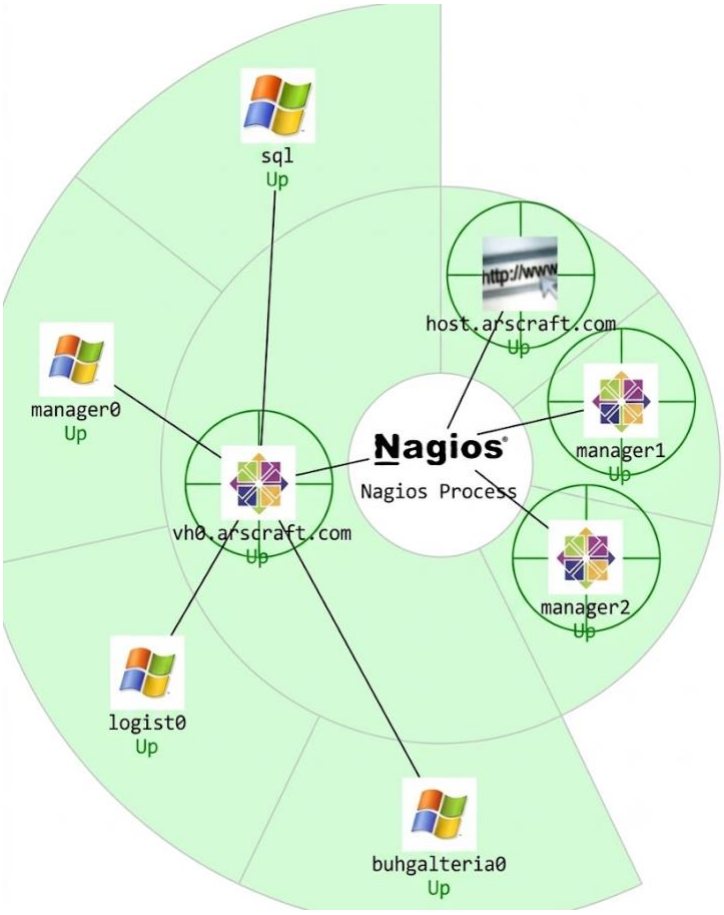


Рисунок 3.2 - Представлення пункту меню Map

sql-1c	C:\ Drive Space	OK	06-12-2026 01:13:53	1d 4h 12m 17s	c: - total: 80.01 Gb - used: 21.01 Gb (26%) -
	CPU Load	OK	06-12-2026 01:15:02	0d 11h 41m 8s	CPU Load 0% (5 min average)
	Memory Usage	OK	06-12-2026 01:14:15	24d 9h 40m 4s	Memory usage: total:17212.59 Mb - used: 15
	Ragent	OK	06-12-2026 01:07:19	4d 1h 48m 51s	ragent.exe: Running
	SQL-Server	OK	06-12-2026 01:06:28	4d 9h 45m 17s	sqlserver.exe: Running
	Uptime	OK	06-12-2026 01:09:37	8d 2h 6m 33s	System Uptime - 0 day(s) 11 hour(s) 36 minu
vh0.arscraft.com	Current Load	OK	06-12-2026 01:11:59	31d 10h 3m 8s	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	06-12-2026 01:13:08	31d 10h 31 29s	USERS OK - 1 users currently logged in
	HTTP	WARNING	06-12-2026 01:14:17	31d 10h 31 31s	HTTP WARNING: HTTP/1.1 403 Forbidden -
	PING	OK	06-12-2026 01:15:25	25d 8h 4m 29s	PING OK - Packet loss = 0%, RTA = 0.09 ms
	Root Partition	OK	06-12-2026 01:11:34	21d 8h 4m 29s	DISK OK - free space: / 66387 MB (95% inod
	SSH	OK	06-12-2026 01:12:43	31d 10h 31 35s	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Swap Usage	OK	06-12-2026 01:12:06	31d 1h 3m 38s	SWAP OK - 100% free (509 MB out of 509 M
	Total Processes	OK	06-12-2026 01:13:14	31d 1h 3m 37s	PROCS OK: 27 processes with STATE = RSZ

Рисунок 3.3 - Представлення пункту меню Services

Система моніторингу Nagios використовує набір конфігураційних файлів для визначення того, як і що саме відстежувати. Ці файли, що знаходяться в директорії /usr/local/nagios/etc, дозволяють налаштувати методи перевірки та об'єкти моніторингу [10].

Давайте розглянемо їх призначення:

1. nagios.cfg - це основний файл налаштувань. Він вказує розташування інших конфігураційних файлів, визначає загальні змінні, налаштовує ведення журналів (логів) та визначає обліковий запис, під яким працюватиме Nagios.
2. commands.cfg: у цьому файлі визначаються команди, які Nagios використовує для перевірки стану серверів, таких як їхня доступність або запущені служби.
3. contacts.cfg: тут реєструються контактні дані, на які будуть надсилатися повідомлення про виявлені проблеми з серверами.
4. groups.cfg: цей файл групує сервери та визначає, які саме сервери належать до певних груп.
5. hosts.cfg: тут створюються шаблони для опису обладнання. Ці шаблони визначають параметри перевірки, наприклад, як часто опитувати сервери та коли надсилати попередження.
6. WindowsServer.cfg: це приклад конфігурації для моніторингу серверів під управлінням операційної системи Windows.
7. LinuxServer.cfg: це приклад конфігурації для моніторингу серверів під управлінням операційної системи Linux.

Для збору інформації про стан серверів Nagios використовує протокол SNMP (Simple Network Management Protocol). Щоб моніторити сервери Windows, додатково потрібна установка програми Nsclient++.

Представимо приклад конфігураційного файлу для Windows-сервера:

```
define host
{
# Використовуємо шаблон, створений заздалегідь
use generic-host;
# Ім'я сервера
```

host_name sql;

Аліас імені (псевдонім)

alias sql;

Вказуємо наявність батьківського сервера для формування ієрархічної структури мережі:

parents vh0.arscraft.com ;

Вказуємо IP-адресу хоста

address 172.16.100.34;

Налаштовуємо візуалізацію іконки для веб-інтерфейсу (необов'язковий параметр):

icon_image sql.jpg
statusmap_image sql.jpg

У цьому полі можна вказати додаткові відомості про сервер:

notes_url http://vh0.arscraft.ru/nagios/notes/info-sql.txt
}

Запускаємо моніторинг ключових сервісів. Для SQL-сервера це включатиме відстеження їхньої працездатності та завантаження процесора.

Перевіряємо процес SQL

define service{
use generic-service
host_name sql
service_description SQL-Server

```
check_command check_nt!PROCSTATE!-d SHOWALL -l sqlservr.exe
}
```

Перевіряємо навантаження на процесор

```
define service{
use generic-service
host_name sql
service_description CPU Load
check_command check_nt!CPULOAD!-l 5,80,90}
```

Додавання Linux-серверів відбувається аналогічно Windows. Після додавання файлів серверів для моніторингу та перевірки конфігураційних файлів (за допомогою відповідної команди), можна переглянути результати на сайті. Та при відсутності помилок перезавантажуємо Nagios:

```
Prov> usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Prov> service nagios reload
```

Основне налаштування завершено, але система дозволяє подальшу кастомізацію. Сповіщення можна гнучко налаштовувати для кожного сервера та процесу, отримуючи листи на пошту системного адміністратора при спрацьовуванні заданих умов.

```
***** Nagios *****
Notification Type: PROBLEM
Host: sql
State: DOWN
Address: 172.16.100.34
Info: (Host Check Timed Out)
Date/Time: Tue Apr 12 09:43:56 MSD 2011
```

Після впровадження та конфігурації системи Nagios, корпоративна мережа отримує розширені можливості для спостереження та контролю [10]:

– контроль за роботою ключових мережевих сервісів: система дозволяє відстежувати працездатність таких служб, як електронна пошта (SMTP, POP3), веб-сервери (HTTP), групи новин (NNTP), а також перевіряти доступність пристроїв за допомогою протоколу ICMP та інших;

– оцінка стану комп'ютерів у мережі: Nagios може аналізувати ключові показники продуктивності серверів, включаючи завантаження процесора, використання дискового простору та системні журнали, для більшості поширених операційних систем;

– гнучкість у створенні власних інструментів моніторингу: завдяки простій архітектурі розширень (плагінів), є можливість розробляти власні методи перевірки служб, використовуючи будь-яку зручну мову програмування, наприклад, Shell, C++, Perl, Python, PHP, C# тощо;

– ефективне паралельне сканування: система здатна одночасно перевіряти стан багатьох служб, що значно прискорює процес моніторингу;

– визначення залежностей між пристроями: можливість встановлювати зв'язки між хостами ("батьківські" та "дочірні") допомагає точно ідентифікувати причину недоступності мережевих ресурсів, розрізняючи справжні збої від тимчасової відсутності зв'язку;

– оперативне оповіщення про проблеми: у разі виникнення збоїв у роботі служб або пристроїв, система автоматично надсилає сповіщення через електронну пошту, SMS або інші канали, налаштовані користувачем;

– автоматизація реагування на інциденти: можна налаштувати автоматичні дії (обробники подій) для служб або хостів, що дозволяє проактивно вирішувати потенційні проблеми;

– ефективне управління логами: система забезпечує автоматичну ротацію лог-файлів, що полегшує їх аналіз та зберігання;

– побудова масштабованих рішень: дозволяє об'єднувати кілька систем моніторингу для підвищення надійності та створення розподіленої інфраструктури спостереження;

– зручна звітність: вбудована утиліта nagiostats надає комплексний звіт про стан усіх моніторингових об'єктів.

Переваги використання Nagios для контролю корпоративної мережі очевидні. Системний адміністратор завжди буде в курсі актуального стану серверів, що дозволить йому оперативно реагувати на будь-які проблеми,

попереджати їх або швидко усувати. Процес встановлення та налаштування системи не повинен викликати значних труднощів у обслуговуючого персоналу.

3.2 Система моніторингу продуктивності серверів Cacti

Cacti - це веб-додаток з відкритим кодом, призначений для візуалізації даних за допомогою інструментарію RRDtool. Система забезпечує збір статистичних показників через визначені часові інтервали та їх подальше графічне представлення. Типові сценарії використання передбачають застосування стандартних шаблонів для моніторингу завантаження процесора, використання оперативної пам'яті, кількості активних процесів, дискового простору, а також вхідного та вихідного мережевого трафіку [10].

Завдяки функціоналу Cacti можливо проводити детальний аналіз пікових навантажень на сервери, а також відстежувати динаміку споживання ресурсів у розрізі дня, тижня чи місяця. Результати аналізу отриманих графіків дозволяють обґрунтувати потребу в модернізації серверного обладнання або оптимізації процесів у періоди максимального навантаження [10].

Процес інсталяції та налаштування системи буде виконано на маршрутизаторі Prov. З огляду на те, що Cacti потребує встановлення низки додаткових програмних компонентів, переходимо до етапу інсталяції, враховуючи, що наявний веб-сервер уже налаштований.

Перелік необхідного програмного забезпечення: MySQL, PHP, Perl та відповідні бібліотеки.

```
Prov> yum install mysql mysql-server mysql-devel httpd httpd-devel php
php-mysql php-gd phpimap
php-ldap php-odbc php-pear php-xml php-xmlrpc php-mcrypt curl curl-
devel perl-lib libxml2 php-mbstring phpmyadmin
```

Запускаємо та конфігуруємо mysql.

```
Prov> service mysqld start
```

Переходимо до встановлення Cacti. Встановлюємо у автозавантаження та запускаємо сервіс SNMP. Також завантажуємо та розпаковуємо пакети Cacti

```
Prov> yum install -y net-snmp net-snmp-utils rrdtool php-snmp
Prov> chkconfig snmpd on
Prov> service snmpd start
Prov> wget www.cacti.net/downloads/cacti-0.8.7g.tar.gz
Prov> wget www.cacti.net/downloads/pia/cacti-plugin-0.8.7g-PA-v2.9.tar.gz
Prov> tar -xzvf cacti-0.8.7g.tar.gz
Prov> tar -xzvf cacti-plugin-0.8.7g-PA-v2.9.tar.gz
```

Після цього створюємо робочу папку Cacti на сервері та користувача з необхідними правами.

```
Prov> mkdir /var/www/cacti
Prov> cp -rf cacti-0.8.7g/* /var/www/cacti/
Prov> useradd -c CactiUser -d /var/www/cacti/ -s /sbin/nologin cactiuser
Prov> chown -R cactiuser /var/www/cacti/log /var/www/cacti/rra
```

Створюємо базу даних та вводимо привілеї для створеного користувача cactiuser:

```
Prov> mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 55
Server version: 5.0.77 Source distribution
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> create database cacti;
mysql> GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY
'password';
mysql> flush privileges;
mysql> quit
```

Імпортуємо сформовану структуру Cacti в базу та налаштовуємо доступ до бази даних

```
Prov> mysql -u cactiuser -p cacti < /var/www/cacti/cacti.sql
Enter password: (password)
Prov> nano /var/www/cacti/include/config.php
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "password";
$database_port = "3306";
```

Відкриваємо робочу директорію Cacti та встановлюємо офіційні патчі

```
Prov> cd /var/www/cacti
Prov> wget
www.cacti.net/downloads/patches/0.8.7g/data_source_deactivate.patch
Prov> wget www.cacti.net/downloads/patches/0.8.7g/graph_list_view.patch
Prov> wget www.cacti.net/downloads/patches/0.8.7g/html_output.patch
Prov> wget
www.cacti.net/downloads/patches/0.8.7g/ldap_group_authenication.patch
Prov> wget
www.cacti.net/downloads/patches/0.8.7g/script_server_command_line_pars
e.patch
Prov> wget www.cacti.net/downloads/patches/0.8.7g/ping.patch
Prov> wget www.cacti.net/downloads/patches/0.8.7g/poller_interval.patch
Prov> patch -p1 -N < data_source_deactivate.patch
Prov> patch -p1 -N < graph_list_view.patch
Prov> patch -p1 -N < html_output.patch
Prov> patch -p1 -N < ldap_group_authenication.patch
Prov> patch -p1 -N < script_server_command_line_parse.patch
Prov> patch -p1 -N < ping.patch
Prov> patch -p1 -N < poller_interval.patch
```

Для включення веб-доступу створюємо файл cacti.conf:

```
Prov> nano /etc/httpd/conf.d/cacti.conf
Alias /cacti /var/www/cacti
<Directory /var/www/cacti/>
DirectoryIndex index.php
Options -Indexes
AllowOverride all
order deny,allow
deny from all
allow from 192.168.1.0/24 (можна зробити так: "allow from all")
AddType application/x-httpd-php .php
php_flag magic_quotes_gpc on
php_flag track_vars on
</Directory>
# Перезавантажуємо веб-сервер Apache
Prov> service httpd restart
# Створюємо завдання Cron для Cacti
Prov> nano /etc/cron.d/cacti
*/5 * * * * cactiuser php /var/www/cacti/poller.php > /dev/null 2>&1
```

Після завершення базового налаштування Cacti можна розпочати роботу з системою. Додавання пристроїв для моніторингу здійснюється через веб-інтерфейс, доступний за адресою: http://ip.адрес.сервера/cacti. Попередньо на цільових серверах необхідно встановити пакет SNMP та оновити конфігураційний файл /etc/snmp/snmpd.conf.

```
syslocation Test.syscontact INF <sysadmin@arscraft.ru>
# sec.name source community (password)
com2sec Mybox localhost public
com2sec cacti ip.адреса.сервера public
com2sec Outside default public
# group.name sec.model sec.name
```

```
group RWGroup v2c Mybox
group ROGroup v1 cacti
group ROGroup v2c cacti
group Others v2c Outside
view all included .1 80
view system included system fe
# context sec.model sec.level prefix read write notif
access ROGroup "" any noauth exact all none none
access RWGroup "" v2c noauth exact all all all
access Others "" v2c noauth exact system none all
```

Приклади графіків що отримані від системи моніторингу приведено на рисунках 3.4, 3.5, 3.6, 3.7.

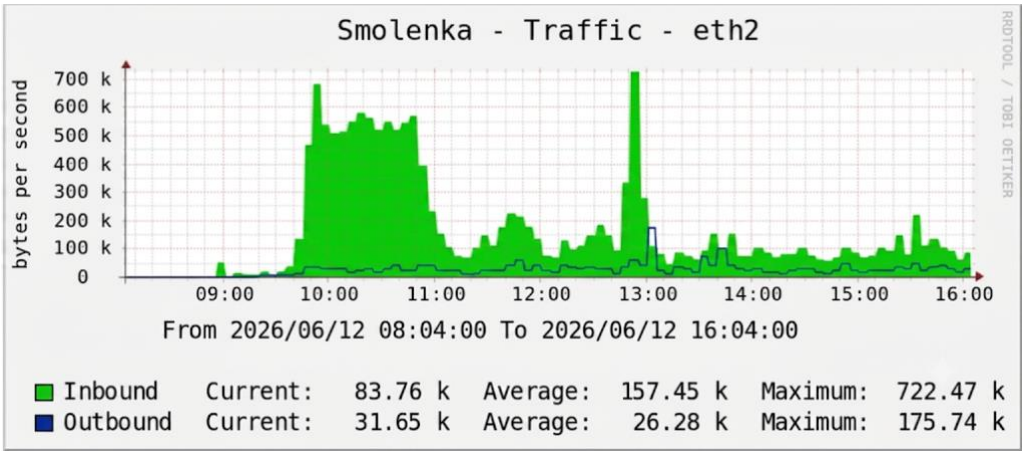


Рисунок 3.4 - Графік використання Інтернет-каналу

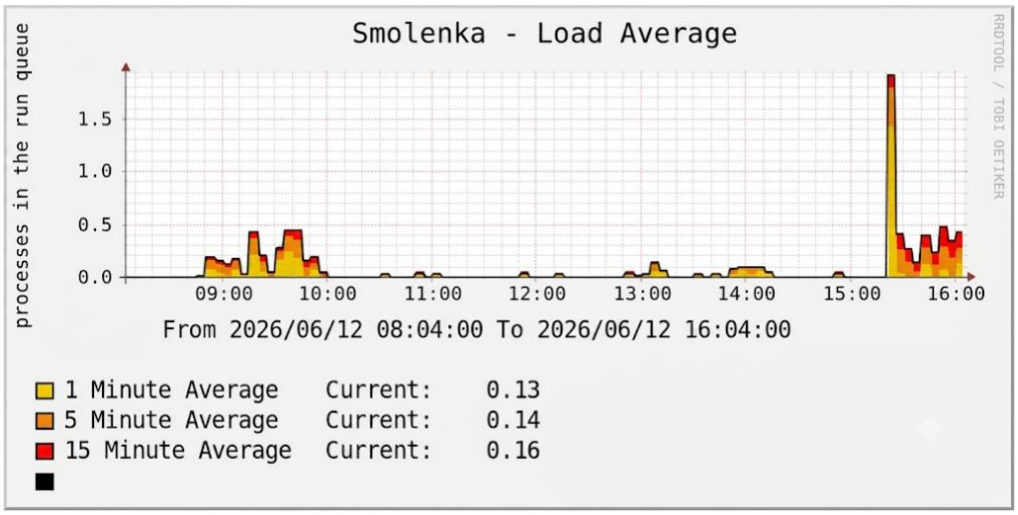


Рисунок 3.5 - Графік середнього завантаження процесора

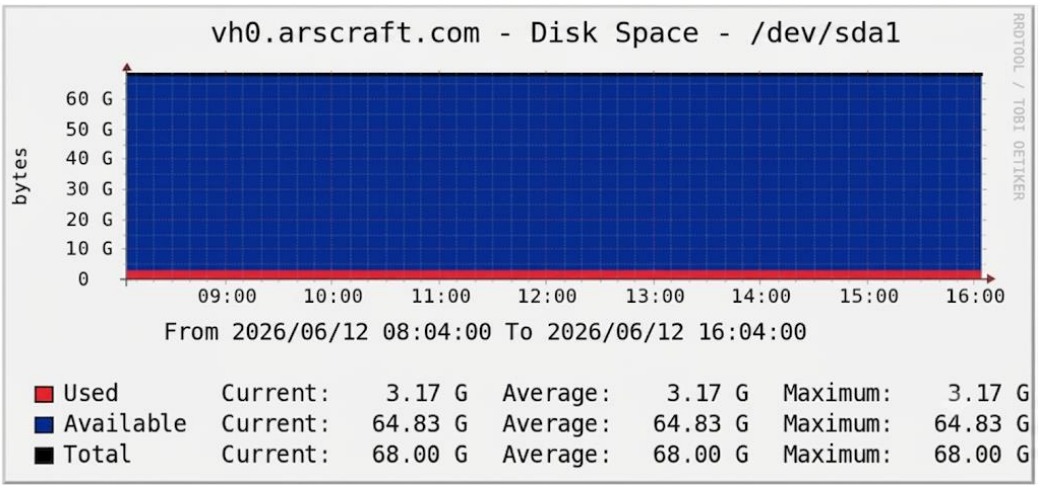


Рисунок 3.6 - Графік використання дискового простору

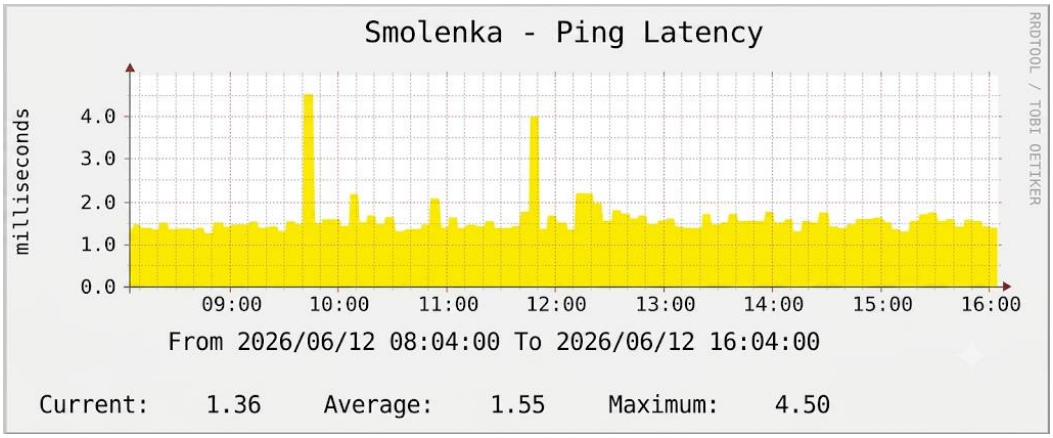


Рисунок 3.7 - Графік відгуку сервера (RTT)

3.3 Методи аналізу трафіку в корпоративній мережі

3.3.1 Використання Proxy-серверу

У корпоративному середовищі, де співробітники мають доступ до мережі Інтернет, впровадження фільтрації вхідного трафіку є критично важливим завданням. Це зумовлено двома основними чинниками:

1. Кібербезпека: більшість випадків інфікування робочих станцій шкідливим програмним забезпеченням пов’язана з відвідуванням вебресурсів сумнівної репутації, переглядом інфікованих банерів та завантаженням неперевіреного ПЗ.

2. Оптимізація робочого часу: однією з типових вимог керівництва є регламентація доступу до розважальних ресурсів та соціальних мереж для підвищення продуктивності праці.

Організація процесу фільтрації вимагає спрямування всього HTTP-трафіку через проксі-сервер.

Проксі-сервер - це мережевий вузол, що виступає посередником у взаємодії між клієнтом та цільовими серверами. Клієнт надсилає запит до проксі-сервера, який, своєю чергою, звертається до зовнішнього ресурсу, отримує дані та передає їх користувачу. За наявності кешування проксі-сервер може видавати дані безпосередньо з локального сховища, що підвищує швидкість роботи та зменшує навантаження на зовнішній канал. Крім того, проксі-сервер дозволяє модифікувати запити/відповіді, забезпечувати захист від певних типів мережевих атак та підвищувати рівень анонімності клієнтів.

Існує два основні методи перенаправлення трафіку на проксі-сервер:

1. Явне налаштування: ручне зазначення адреси проксі-сервера в налаштуваннях браузера (метод є неефективним через складність підтримки відмовостійкості).

2. Прозоре проксіювання: автоматичне перенаправлення трафіку, що надходить на порти 80 та 8080 маршрутизатора, за допомогою правил Iptables.

У межах кваліфікаційної роботи буде реалізовано другий метод із використанням програмного забезпечення Squid, що встановлюється на маршрутизатор Prov.

Squid - це кешувальний проксі-сервер з відкритим вихідним кодом (ліцензія GNU GPL), який забезпечує підтримку протоколів HTTP, FTP, Gopher та HTTPS (за умови належної конфігурації). Програма функціонує як єдиний неблокуючий процес введення/виведення, що забезпечує високу ефективність обробки запитів.

Встановлення пакета здійснюється зі стандартного репозиторію за допомогою наступної команди:

Prov> yum install squid

Ініціалізовуємо кеш:

Prov> squid -z

Переходимо до етапу налаштування. Необхідно реалізувати обмеження

доступу до розважальних ресурсів та соціальних мереж, а також заборонити завантаження файлів із розширеннями *.exe, *.sys, *.bat, *.mp3, *.avi, *.mp4, *.mov.

Основна конфігурація проксі-сервера здійснюється у файлі /etc/squid/squid.conf. Приведено необхідні зміни.

Вказуємо порт для прийому запитів та встановлюємо тип проксі-сервера як «прозорий» (transparent), оскільки пересилання пакетів здійснюється засобами маршрутизатора:

```
http_port 3128 transparent
```

ACL (Access Control List - список контролю доступу) визначає права доступу суб'єктів до об'єктів мережі та дозволені типи операцій. Використовуючи директиву acl, описуємо відповідні об'єкти та формуємо правила доступу до них.

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443
acl Safe_ports порт 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports порт 210 # wais
acl Safe_ports порт 1025-65535 # незареєстровані порти
acl Safe_ports порт 280 # http-mgmt
acl Safe_ports порт 488 # gss-http
acl Safe_ports порт 591 # filemaker
acl Safe_ports порт 777 # multiling http
acl CONNECT метод CONNECT
http_access allow manager localhost
http_access deny manager
```

```
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

Описуємо корпоративну мережу

acl localnet src 172.16.100.0/24
# Визначення переліку заборонених розширень файлів
acl banfiles urlpath_regex -i "/etc/squid/include/permblock.files.acl"
# Визначення переліку заборонених доменів
acl banurl dstdomain -i "/etc/squid/include/permblock.url.acl"
# Визначення правил для обмеження аудіо- та відеоконтенту на основі MIME-типів
acl banvideo rep_mime_type content-type audio
acl banvideo rep_mime_type content-type video
# Застосування обмежень доступу для локальної мережі
http_access deny localnet banfiles
http_access deny localnet banurl
http_reply_access deny all banvideo
# Надання дозволу на решту трафіку
http_access allow localnet
# Заборона всього для для всіх інших
http_access deny all
```

Необхідно забезпечити надання адміністраторських прав шляхом додавання відповідного запису ACL. Розглянемо файли, що містять списки заборонених об'єктів.

```
Permblock.files.acl:
\.exe$
\.msi$
\.bat$
\.sys$
Permblock.url.acl:
.youtube
```

…

Наступним етапом є перенаправлення запитів із портів 80 та 8080 на порт 3128. Для реалізації цього завдання до ланцюжка iptables необхідно додати відповідне правило:

$fw -t nat -A PREROUTING -p tcp -m multiport --dport 80,8080 -i eth1 -j DNAT --to 127.0.0.1:3128

Після виконання цієї операції трафік спрямовуватиметься через проксі-сервер для подальшої фільтрації. Функціонал проксі-сервера також дозволяє встановлювати ліміти на споживання трафіку, налаштовувати роботу за розкладом та здійснювати заміну рекламних банерів на вебсторінках.

3.3.2 Настроювання аналізатора логів проксі-сервера

Аналіз мережевого трафіку є одним із ключових завдань, що виконується з метою:

1. Моніторингу активності персоналу для забезпечення дотримання корпоративної дисципліни та внутрішніх регламентів.

2. Оптимізації витрат компанії шляхом встановлення лімітів на використання трафіку.

Проксі-сервер Squid автоматично фіксує всі параметри з’єднань (джерело запиту, цільовий ресурс, часові мітки, обсяг даних тощо) у лог-файлі. Для отримання аналітичних даних необхідно обробити ці записи та візуалізувати їх у зручному для сприйняття форматі. Нижче наведено приклад логу проксі-сервера Squid:

1305189448.705 1001 195.239.137.178 TCP_MISS/200 1363 POST http://mail.google.com/a/shop.com.ua/? shop1 DIRECT/74.125.79.18 text/javascript
1305189550.826 243750 195.239.137.178 TCP_MISS/200 638 GET http://mail.google.com/a/shop.com.ua/channel/bind? shop1 DIRECT/74.125.79.18 text/plain

Для аналізу трафіку буде використано спеціалізоване програмне забезпечення з відкритим кодом - LightSquid.

Оскільки всі необхідні залежності (Perl, Gd, httpd) уже налаштовані, переходимо безпосередньо до процедури інсталяції:

1. Завантажуємо останню версію програмного забезпечення (v1.8) з офіційного ресурсу: http://lightsquid.sourceforge.net/.

2. Розпаковуємо архів у директорію /var/www/html/ та вносимо необхідні зміни

```
Prov> tar -xzf lightsquid-1.8.tgz
Prov> cd lightsquid-1.8
# Надаємо права на виконання скриптів
Prov> chmod +x *.cgi
Prov> chmod +x *.pl
Prov> chown -R apache:apache *
Alias /lightsquid "/usr/local/www/lightsquid-1.8"
<Directory "/var/www/html/lightsquid-1.8">
AddHandler cgi-script .cgi
AllowOverride All
</Directory>
```

Для застосування внесених змін необхідно перезавантажити веб-сервер

```
Prov> service httpd restart
Prov> cd /var/www/html/lightsquid-1.8
```

Виконуємо перевірку конфігураційного файлу на наявність помилок. У разі їх виникнення, ймовірною причиною є некоректно вказані шляхи до робочих каталогів програми. Коригування параметрів здійснюється у файлі lightsquid.cfg.

```
Prov>./check-setup.pl
```

За відсутності помилок необхідно додати завдання з аналізу трафіку до

планувальника cron для виконання з періодичністю 30 хвилин:

```
Prov> crontab –e
*/30 * * * * /var/www/html/lightsquid-1.8/lightparser.pl
```

На цьому етапі процес інсталяції завершено. За потреби можна розподілити IP-адреси джерел трафіку за групами та надати їм відповідні найменування.

Результати роботи системи доступні за посиланням: http://ip.адреса.сервера/lightsquid. Приклади функціонування аналізатора трафіку наведено на рисунках 3.8, 3.9, 3.10.

Звіт з використання інтернету, проксі-сервер Squid.

Дата: 06 червня 2026 (Оновлено :: 11:30 :: 12 червня 2026)

Популярні сайти (звіт)

Хто завантажив ВЕЛИКІ файли (звіт)

№	Час	Користувач	П.І.Б.	З'єднань	Байт	%	Група
1		office1	office1 user name	20 558	150.8 M	90.2%	01. office1
2		M1	M1 user name	1 880	12.0 M	7.1%	03. M1
3		M2	M2 user name	74	4.3 M	2.5%	02. M2

Рисунок 3.8 - Приклад звіту LightSquid

Всього **150.8 M**

№	Відвідані сайти	З'єднань	Байт	Разом	%
1	www.gmail.com	4 181	36.2 M	36.2 M	24.0%
2	www.prom.ua	1 197	12.5 M	48.7 M	8.2%
3	www.google.com	374	7.0 M	55.6 M	4.6%
4	www.rozetka.com.ua	151	4.6 M	60.3 M	3.0%
5	www.web.telegram.org	262	3.7 M	64.0 M	2.4%

Рисунок 3.9 - Відвідані сайти

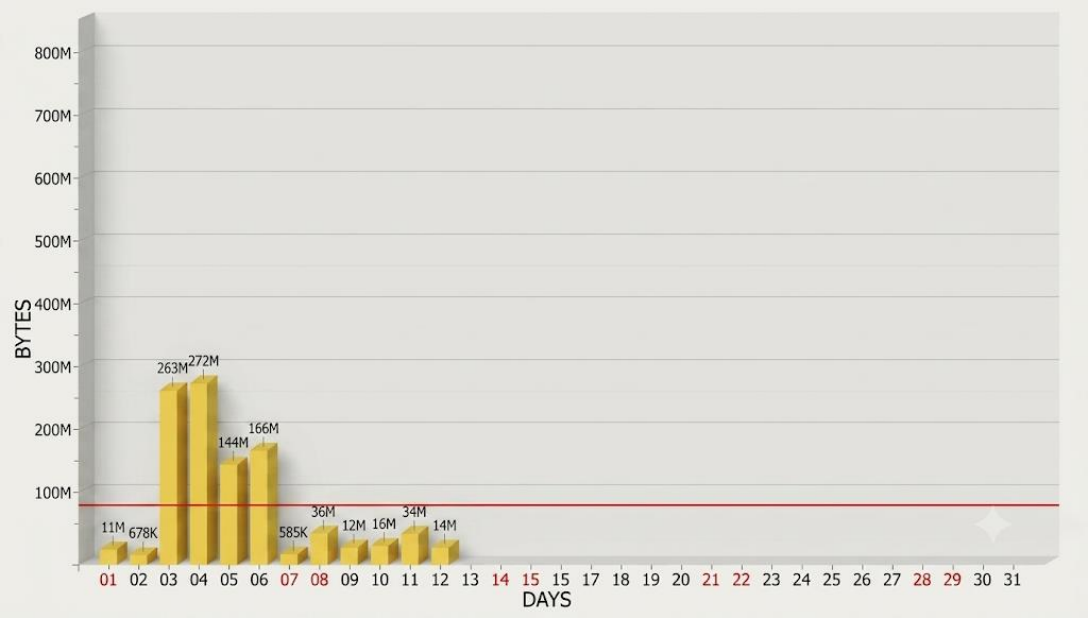


Рисунок 3.10 - Графік використання трафіку за днями

3.4 Аналіз трафіку корпоративної мережі

Моніторинг активності співробітників засобами аналізу HTTP-трафіку дозволяє вирішити поставлене завдання лише частково, оскільки не враховує інші протоколи передачі даних.

Багато інтернет-провайдерів застосовують тарифікацію за обсягом спожитого трафіку для фізичних та юридичних осіб. Перевищення встановлених лімітів призводить до значних додаткових витрат, що зумовлює необхідність впровадження ефективної системи моніторингу трафіку.

На ринку представлено чимало білінгових систем, проте більшість із них є надлишковими для вирішення даного локального завдання через велику кількість нерелевантного функціоналу. Оптимальним інструментом у такому разі є програмне забезпечення IPCad.

IPCad (Cisco IP accounting simulator) - це спеціалізоване рішення для обліку трафіку, що підтримує роботу через інтерфейси BPF, libpcap та iptables. Однак, програмний продукт має певні обмеження: вивід даних здійснюється безпосередньо в консоль без попередньої фільтрації, що потребує доопрацювання інтерфейсу взаємодії.

Процес встановлення IPCad розпочинається з наступних дій:

router> wget http://lionet.info/soft/ipcad-3.6.6.tar.gz

Після завантаження необхідно розпакувати архів та перейти до відповідного каталогу.

```
router>tar -xvzf ipcad-3.6.6.tar.gz
router>cd ipcad-3.6.6
```

Налаштовуємо та встановлюємо пакет

```
router>./configure
router>make
router>make install
```

Формуємо файл конфігурації під свої потреби

```
router> nano /usr/local/etc/ipcad.conf
capture-ports disable;
interface eth1;
```

Налаштовуємо програми за замовчуванням

```
netflow export version 5;
netflow export destination 127.0.0.1 9998;
pidfile = /var/run/ipcad.pid;
dumpfile = ipcad.dump;
buffers = 64k;
```

Виконуємо розподіл статистики для кожної IP-адреси в межах підмережі 192.168.23.0/24.

Параметр «aggregate 192.168.23.0/24» визначає діапазон адрес мережі для ipcad. Параметр «strip 32» забезпечує фіксацію всіх 32 бітів адреси в межах вказаного діапазону при зборі статистичних даних.

Конфігураційний запис:

aggregate 192.168.23.0/24 strip 32;

Визначення політик доступу до статистики ipcad: користувач root має повні права доступу (резервне копіювання, перегляд та редагування таблиць обліку). Для інших користувачів доступ обмежується виключно переглядом статистичних даних.

rsh enable at 127.0.0.1;
rsh root@127.0.0.1 admin;
rsh root@127.0.0.1 backup;
rsh root@127.0.0.1;
rsh 127.0.0.1 view-only;

Встановлюємо тайм-аут IP-пакета.

rsh ttl = 3;
rsh timeout = 30;

Параметр `memory_limit` визначає обсяг оперативної пам'яті, виділений для зберігання вмісту одного потоку даних.

memory_limit = 10m;

Для отримання статистичних даних виконуємо такі команди:

Prov> /usr/local/bin/ipcad -rds
Prov> rsh localhost show ip accounting

У результаті виконання в консолі відобразяться відповідні відомості:

192.168.23.14 83.156.177.10 1 131
178.67.60.119 192.168.23.14 1 305
192.168.23.14 202.152.243.92 1 131
192.168.23.14 178.67.60.119 1 131

192.168.23.14 77.77.44.16 1 131
192.168.23.196 192.168.23.201 2873 186687
192.168.23.201 192.168.23.196 4274 3838143

Наведені результати мають технічний характер і потребують подальшої автоматизації для ефективного моніторингу трафіку. Ручне виконання команд є неефективним методом збору даних, тому доцільним рішенням є розробка спеціалізованого веб-інтерфейсу.

Для реалізації цього завдання необхідно інтегрувати збір статистичних даних із системою керування базами даних MySQL. Процес передбачає створення бази даних stat та відповідних таблиць (users, download, upload, tmp) для систематизації інформації.

Технічна реалізація включатиме:

1. Створення скрипта stat.sh, призначеного для виконання запитів на отримання статистики та виведення результатів у проміжний файл.

2. Розробку скрипта collect.pl, який забезпечуватиме парсинг даних із файлу та їх подальше занесення до бази даних.

Скрипт stats.sh:

```
#!/bin/bash
rsh 127.0.0.1 clear ip accounting > /dev/null
rsh 127.0.0.1 show ip accounting checkpoint > /tmp/ipcad.stat
/usr/local/etc/collect.pl
rm /tmp/ipcad.stat
```

Частина скрипта collect.pl:

```
#!/usr/bin/perl
use DBI;
#Логін / пароль
$dbuser = "root";
$dbpassword = "password";
dbh = DBI->connect("DBI:mysql:stat:localhost",$dbuser,$dbpassword) or
die "неможливо підключитися до бази даних", $dbh->errstr, __LINE__;
open(FIL, "/tmp/ipcad.stat") || die;
while($line=<FIL>) {
```

```
@mass = split(" ",$line);
$dbh -> do ("INSERT INTO `tmp` ( `id` , `date` , `time` , `src` , `srcp` , `dst`
, `dstp` , `bytes` , `proto`,`class` ) VALUES (", CURDATE( ) , CURTIME(
)$
}
close(FIL);
```

Для автоматизації виконання скрипта stat.sh з періодичністю 20–25 хвилин необхідно додати відповідне завдання до планувальника cron:

```
router> crontab –e
0,20,40,55 * * * * /usr/local/etc/stat
```

Після успішного накопичення даних у базі необхідно забезпечити їх відображення на веб-інтерфейсі. Для цього розроблено набір сторінок, зокрема index.php, код якої наведено у додатку В. Сформована система дозволяє моніторити трафік у форматі структурованої таблиці (рис. 3.11) та надає такі функціональні можливості:

- індивідуальний облік споживання трафіку для кожного користувача;
- встановлення лімітів трафіку з автоматичним надсиланням сповіщень на електронну пошту системного адміністратора у разі їх перевищення.

IP	Downloads(MB)	Uploads(MB)	Online
192.168.23.40	53	17	●
192.168.23.38	63	8	●
192.168.23.41	67	21	●
192.168.23.234	492	17	●
192.168.23.32	151	10	●
192.168.1.255	0	0	●
192.168.23.14	44	10	●
192.168.23.174	274	58	●
192.168.23.194	494	21	●
192.168.23.17	2	0	●
192.168.23.16	53	5	●
192.168.23.13	14	5	●
192.168.23.12	48	14	●
192.168.23.15	61	7	●
192.168.23.39	100	8	●
192.168.23.183	27	2	●
192.168.23.217	1289	32	●

Рисунок 3.11 - Таблиця використання інтернет-трафіку

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи можна зробити такі висновки:

1. Спроектовано та побудовано захищену корпоративну мережу на базі технологій OpenVPN та SSH.

2. Проведено аналіз методів організації VPN-мереж, зокрема архітектур клієнт-сервер, тунелювання та «точка-точка».

3. Проведено порівняльний аналіз пропускної здатності захищених каналів зв’язку для технологій OpenVPN та SSH. За результатами оцінки продуктивності встановлено, що оптимальним підходом для корпоративного сегмента є комбіноване використання технологій: OpenVPN для побудови захищених мережевих сегментів та SSH - для створення тунелів і віддаленого адміністрування.

4. Здійснено огляд та обґрунтування вибору рішень для моніторингу корпоративної мережі (Nagios, Cacti, Ipcad, LightSquid) з акцентом на їх комплексне використання.

5. Забезпечено фільтрацію трафіку та контроль активності персоналу шляхом впровадження та налаштування проксі-сервера.

6. Інтегровано та введено в експлуатацію комплексну систему моніторингу на базі програмних продуктів Nagios, Cacti, Ipcad та LightSquid.

7. Розроблено web-інтерфейс для централізованої системи обліку мережевого трафіку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Richard A. Deal Cisco Router Firewall Security / Richard A. Deal, 2009. – 620 с.

2. Що таке VPN-протокол [Електронний документ] - Режим доступу: https://www.cactusvpn.com/ru/beginners-guide-to-vpn/vpn-protocol/ (дата звернення 20.06.2026)

3. Протоколи VPN [Електронний документ] - Режим доступу: https://www.safervpn.com/ru/vpn-protocols (дата звернення 20.06.2026)

4. Пояснення VPN-протоколів: типи, відмінності та як вибрати [Електронний документ] - Режим доступу: https://freevpnplanet.com/ua/blog/vpn-protocols-what-are-they-and-where-they-are-used/ (дата звернення 20.06.2026)

5. Cisco Systems: документация [Електронний ресурс]. Режим доступа: http://www.cisco.com/en/US/docs. (дата звернення 27.06.2026)

6. Олифер В. Искусство оптимизации трафика / В. Олифер, Н. Олифер // Журнал сетевых решений LAN. - 2001. - № 12. - с. 21–26.

7. Rosen, E. Multiprotocol Label Switching Architecture / E. Rosen, A. Viswanathan, R. Callon. - RFC 3031, 2001. - 61 с.

8. Повний гайд по VPN: історія, архітектура, протоколи [Електронний документ] - Режим доступу: https://research.kr-labs.com.ua/vpn-full-guide/ (дата звернення 27.06.2026)

9. Задерейко О.В. Комп’ютерні мережі [Електронний ресурс]: навчальний посібник / О.В. Задерейко, Н.І. Логінова, А.А. Толокнов. – Одеса, 2022. – 249 с.

10. Основи організації мереж Cisco, том 1 [Текст]: Пер. з англ. – К.: Видавничий дім «Вільямс», 2002. – 512 с.